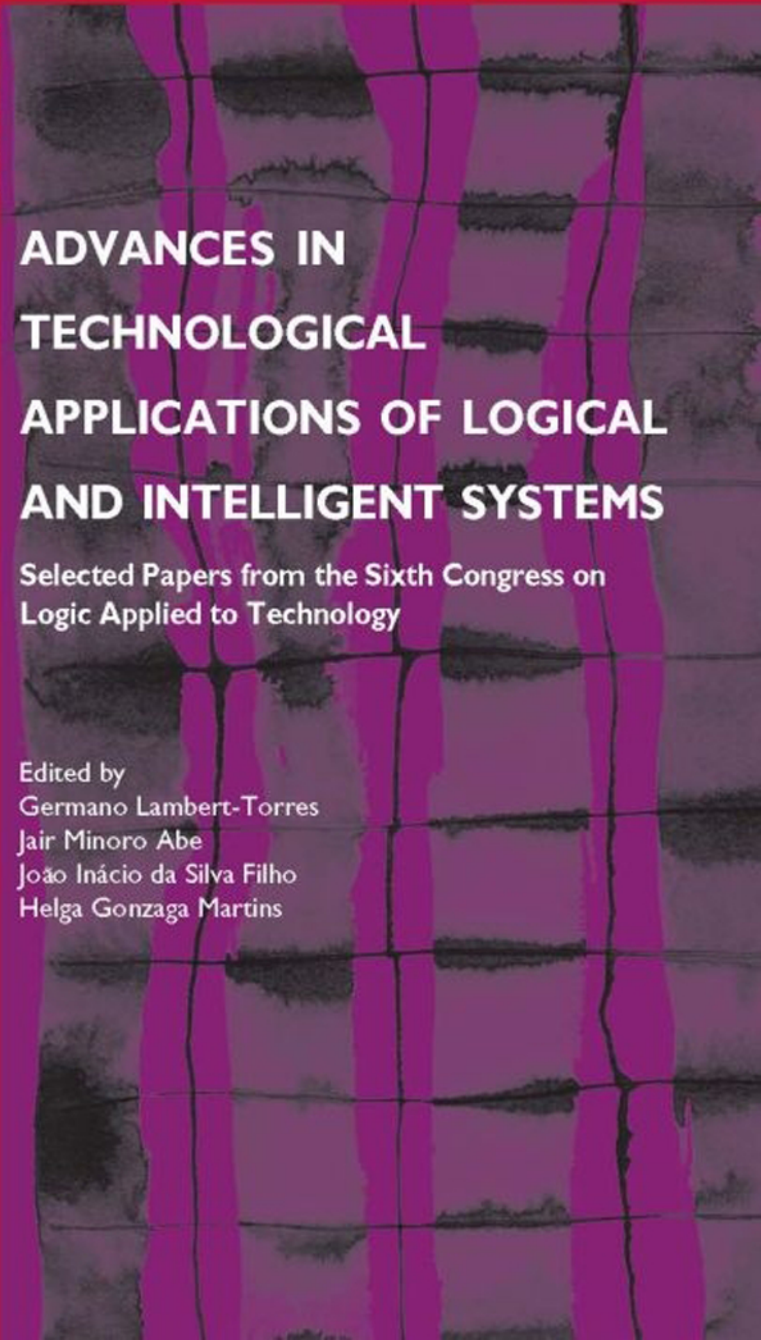
Frontiers
in
Artificial
Intelligence
and
Applications



ADVANCES IN TECHNOLOGICAL APPLICATIONS OF LOGICAL AND INTELLIGENT SYSTEMS

Selected Papers from the Sixth Congress on Logic Applied to Technology

Edited by
Germano Lambert-Torres
Jair Minoro Abe
João Inácio da Silva Filho
Helga Gonzaga Martins

IOS
Press

VISIT…

ADVANCES IN TECHNOLOGICAL APPLICATIONS OF LOGICAL AND INTELLIGENT SYSTEMS

Frontiers in Artificial Intelligence and Applications

Volume 186

Published in the subseries
Knowledge-Based Intelligent Engineering Systems
Editors: L.C. Jain and R.J. Howlett

Recently published in KBIES:

Vol. 180. M. Virvou and T. Nakamura (Eds.), Knowledge-Based Software Engineering – Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering
Vol. 170. J.D. Velásquez and V. Palade, Adaptive Web Sites – A Knowledge Extraction from Web Data Approach
Vol. 149. X.F. Zha and R.J. Howlett (Eds.), Integrated Intelligent Systems for Engineering Design
Vol. 132. K. Nakamatsu and J.M. Abe (Eds.), Advances in Logic Based Intelligent Systems – Selected Papers of LAPTEC 2005

Recently published in FAIA:

Vol. 185. A. Biere, M. Heule, H. van Maaren and T. Walsh (Eds.), Handbook of Satisfiability
Vol. 184. T. Alsinet, J. Puyol-Gruart and C. Torras (Eds.), Artificial Intelligence Research and Development – Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence
Vol. 183. C. Eschenbach and M. Grüninger (Eds.), Formal Ontology in Information Systems – Proceedings of the Fifth International Conference (FOIS 2008)
Vol. 182. H. Fujita and I. Zualkernan (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the seventh SoMeT_08
Vol. 181. A. Zgrzywa, K. Choroś and A. Siemiński (Eds.), New Trends in Multimedia and Network Information Systems
Vol. 179. A. Cesta and N. Fakotakis (Eds.), STAIRS 2008 – Proceedings of the Fourth Starting AI Researchers' Symposium
Vol. 178. M. Ghallab et al. (Eds.), ECAI 2008 – 18th European Conference on Artificial Intelligence
Vol. 177. C. Soares et al. (Eds.), Applications of Data Mining in E-Business and Finance
Vol. 176. P. Zaraté et al. (Eds.), Collaborative Decision Making: Perspectives and Challenges

ISSN 0922-6389

Advances in Technological Applications of Logical and Intelligent Systems

Selected Papers from the Sixth Congress on Logic Applied to Technology

Edited by

Germano Lambert-Torres

Itajuba Federal University, UNIFEI, Itajuba, Brazil

Jair Minoro Abe

Paulista University, UNIP, São Paulo, Brazil

João Inácio da Silva Filho

Santa Cecilia University, UNISANTA, Santos, Brazil

and

Helga Gonzaga Martins

Itajuba Federal University, UNIFEI, Itajuba, Brazil

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 978-1-58603-936-3
Library of Congress Control Number: 2008939409

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

LEGAL NOTICE
The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

Preface

Logic began as the science of valid inference and related topics. It gradually underwent profound changes, widening its initial scope and transforming itself into a mathematical discipline. Today it is a basic science, full of significant concepts and involved results (Goedel's theorems, the theory of forcing, forking, the mathematics of Solovay, etc.) but its main value has always been theoretical.

However, in the twentieth century, logic finally found a number of important applications and originated various new areas of research, especially after the development of computing and the progress of the correlated domains of knowledge (artificial intelligence, robotics, automata, logical programming, hyper-computation, etc.). This happened not only in the field of classical logics, but also in the general field of non classical logics. This reveals an interesting trait of the history of logic: despite its theoretical character, it constitutes, at present, an extraordinarily important tool in all domains of knowledge, in the same way as philosophy, mathematics, natural science, the humanities and technology. Moreover, certain new logics were inspired by the needs of specific areas of knowledge, and various new techniques and methods have been created, in part influenced and guided by logical views.

This book contains papers on relevant technological applications of logical methods and some of their extensions, including: annotated logic and expert systems, fuzzy dynamical models, adaptive devices, intelligent automaton vehicles, cellular automata, information systems and temporal logic, paraconsistent robotics, dynamic virtual environments and multiobjective evolutionary search, cable routing problems, and reinforcement of learning. All papers are well summarized in their abstracts.

This collection of papers gives a clear idea of some current applications of logical (and similar) methods to numerous problems, including relevant new concepts and results, in particular those related to paraconsistent logic. It will be of interest to a wide audience: pure logicians, applied logicians, mathematicians, philosophers and engineers.

September, 2008

Newton C.A. da Costa

This page intentionally left blank

Foreword

Description of the real world is one of the most important steps in problem solving. However, this description is often bedevilled by inconsistency, unclear definitions and partial knowledge. When these apply, the use of *Aristotelian Logic* alone is not sufficient to promote correct reasoning and arrive at the best or the most appropriate answer.

For example, in traditional control systems, classical logic is the theoretical functional base. This means that these systems work with a binary structure; true or false. Using classical logic, some simplifications must be made to the reasoning, such as omitting the consideration of inconsistent facts or situations, or summarizing them very roughly.

Nowadays, many researchers have developed new methods which allow for optimum consideration of real situations. The Congress on Logic Applied to Technology (LAPTEC) is held to facilitate the coming together of academics and professionals from industry and to create an environment for the discussion of possible applications of logic and intelligent systems to solve real problems.

Previous LAPTEC congresses have been held in cities in Brazil and Japan. This book contains a set of selected papers published at the 6th Congress on Logic Applied to Technology (LAPTEC) held in Santos, Brazil, from November 21st to 23rd 2007. This congress covered areas such as Artificial Intelligence, Automation and Robotics, Classical and Non-Classical Logics, Computability, Informatics, and Technology. The chapters of this book present applications from different areas such as: industry, control systems, robotics, power systems, and medical decision-making.

The Organization Committee of the LAPTEC would like to express their gratitude to reviewers and session chairs who contributed to the running of the congress; without their work, it would have been impossible for this congress to take place. We would also like to thank Energias do Brasil, CAPES, CNPq, FAPESP, and FAPEMIG for their financial support for the congress. Finally, special thanks go to the Board of Directors of the Santa Cecilia University for providing a venue for the congress and all the facilities which made the event such a brilliant success.

The authors dedicate this book to Professor Newton da Costa, the Father of *Paraconsistent Logic*, author of innumerable significant contributions to the field of mathematics, for his lessons and continuing inspiration.

September, 2008

Germano Lambert-Torres
Jair Minoro Abe
João Inácio da Silva Filho
Helga Gonzaga Martins

This page intentionally left blank

Contents

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-1

Algebraic Framework for Reverse Engineering on Specifications

Isabel CAFEZEIRO [a] and Edward Hermann HAEUSLER [b]
[a] *Departamento de Ciência da Computação, Universidade Federal Fluminense*
Rua Passo da Pátria, 156 - Bloco E - Niterói Brasil CEP: 24.210-240 Email: isabel@dcc.ic.uff.br
[b] *Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)*
Rua Marques de São Vicente 225, Gávea, Rio de Janeiro, RJ 22453-900, Brazil.
Email: hermann@inf.puc-rio.br

Abstract. This paper focuses on the problem of providing a well founded framework to make possible the reverse engineering on algebraic specifications. The paper considers existing and well-accepted formalizations for operations to compose specifications and propose a categorical formalization to operations to break specifications. The concept of categorical limit is adopted, and categorical equalizers are proposed to formalize operations to hide components of specifications. The proposed formalizations complete an algebraic framework that makes possible the manipulation of specifications, composing and decomposing old specifications in order to obtain a better use of existing specification.

keywords. Reverse Engineering, Algebraic Specification, Category Theory

Introduction

Specification is the part of the process of software development that precedes the implementation and succeeds the requirements refinements. As well as other phases of software development, it is developed by a stepwise refinement, and goes from an abstract view of *what* must be implemented to a (possibly executable) description of a system. If the elicitation of software requirements properly reflects the users aspirations, the stepwise refinement of a specification tends to propagate its features towards the final code. In this way, (formal) specification strongly contributes in the construction of reliable systems. In addition, specifications decrease the cost of the system contributing with the detection of misunderstandings in the initial phases of the process, and also make possible the verification of certain systems properties before its implementation.

Algebraic Specifications are those in that the object to be specified is constructed by means of properties expressed by equations. The text of an algebraic specification is composed by functions symbols and axioms (equations where these symbols appear). The axioms bind the meaning of the specification, which turns out to be any object that can be described by means of the function, with the behavior guided by the axioms.

In the 80s concepts of Algebraic Specifications were developed to formalize data types [10,15]. Since then, many algebraic techniques of specification were used in

Computer Science, giving rise to libraries of specifications to be composed in a modular way, forming large systems [4].

As well as in other phases of software construction, *reuse* (and, *modularity* as a premise) is a key word in the process of specification. Within the context of reuse we identify the activities of recognizing, decomposing and classifying specifications as the *Reverse Engineering on Specifications*, which can be ultimately characterized as the inverse process of specification development, that goes from lower to upper levels of abstraction. Reverse Engineering on Specifications brings to the context of specifications the same benefits as re-engineering to software construction [2,3] it improves understanding and readability of specifications giving rise to more reliable software systems.

The process of "re-engineering" specifications can be enormously facilitated by a framework that makes possible to break and compose specifications in several ways, extracting or extending its functionality. Within this context, we can enumerate a small and useful set of operations that are traditionally used to put specifications together, extending its functionality. For example, the operation *enrich* is used to add to a specification components that come from another specification [10,13,15,16]. *Union* is used to add information, possibly collapsing components. In addition, parameterising specifications is also possible. There are also operations that preserve the informative power of specifications. For example, *rename*. Finally, we must consider operations that reduce functionality of specifications producing "smaller" specifications than the original ones. Operations in this set are very important, as they make possible to cut undesirable features and to extract part of the functionality of a specification. They are also useful to extract common meaning of specifications, making possible to re-arrange a library of specifications. In this set, we can cite *derive*, which builds a new specification by hiding part of a specification. It would be useful to have an operation to give the functional intersection of specifications.

These three sets of operations (that increase, preserve or reduce the expressive power of specifications) form a useful framework to help managing libraries of specifications and building large specifications in a modular way, making possible the reverse engineering. But it is essential that the complete framework is well defined and well founded. Traditionally, Category Theory is the formalism used in the community of specifications. It provides elegant and concise definitions for the first two sets of operations cited above. In the case of the third mentioned set, however, the formalization of the operations is very complex and requires categorical concepts that are not familiar to most of the community. Because of this, in spite of its large applicability, formalizations of these operations are not frequently presented.

In [7] we proposed the formalization of an operation to extract common meaning of two or more specifications based in the categorical concept of *limit*. Extending [7], we propose in this paper a categorical formalization to extract part of the functionality of a specification. We show how a particular case of categorical limit can be used to formalize an operation of hiding specification components. We also develop another example to illustrate the applicability of limit as a way of formalizing semantic intersection. With these operations we complete a useful formal framework to manipulate specifications make viable the reverse engineering. We adopt basic categorical constructs preserving the simplicity and the elegance of the traditional formalizations in the area.

This paper is organized as follows: Section 1 presents basic concepts ion algebraic specifications. Section 2 defines the category of algebraic signatures and presents limit concepts. Section 3 illustrates limit concept with an application. Section 4 presents the equalizer as an operation to hide symbols of a specification. Section 5 presents related works and comments that the subject of this paper is part of a more complex project. Section 6 concludes the paper.

1 Concepts and Definitions

We start by defining the Category *Sign* of Algebraic Signatures. It is composed by Algebraic signatures, which are related by signature morphisms. Algebraic Signature determines the vocabulary of the specification. It provides a set of sort names and a family of sets of function (or operation) names, where each function name is defined using the sort names. This family of sets is indexed by the size of rank of function. For example, all unary function names are collected in the same set.

Definition 1. An algebraic signature Σ is a pair <S,F> where S is a set (of sort names) and F is an indexed family of sets (of operation names).

For a given $f:u \rightarrow s \in F^n$, $u \in S^*$, $s \in S$, and |us| (the length of us) is equal to n. We say that $f:u \rightarrow s$ has rank us, and we denote F_{us} the subset of F^n whose members have rank us.

Example: $\Sigma = < \{a,b\},F^2, F^3>$, where $F^2 = \{f:a \rightarrow b, f:b \rightarrow b\}$ and $F^3 = \{f:a \times a \rightarrow b\}$ is an algebraic signature with sorts a and b and operation names $f:a \rightarrow b$, $f:b \rightarrow b$ and $f:a \times a \rightarrow b$. The sets $F^1,F^4,F^5,...$, are empty.

The definition of algebraic signature that we adopt in this paper differs from the traditional approach [10,15] as it organizes the family F of sets by rank length. In this way, a morphism linking two signatures must respect not only the translation of sorts, but also the rank length of operation names. In the traditional approach, the family F is organized by ranks, thus, the same signature would be presented as $\Sigma = < \{a,b\},F_{ab},F_{bb},F_{aab}>$, with $F_{ab} = \{f:a \rightarrow b\}$, $F_{bb}= \{f:b \rightarrow b\}$ and $F_{aab} = \{f:a \times a \rightarrow b\}$. A consequence of the traditional approach is that projections (morphisms that recover a component of the limit) fail to be epimorphisms (analogue to surjections in Set Theory), as it would be expected. Thus, limit, in the traditional approach, has not an intuitive presentation (See [8] for categorical details on this topic).

Algebraic signatures are linked by signature morphisms, whose origin is called domain and target is called codomain. These are homomorphism between algebraic signatures. They are composed by a function σ_S that relates sorts of the domain signatures with sorts of the codomain signature, and a family of functions σ_F, one function for each set of the family of sets.

Definition 2. Given Σ= <S,F> and Σ'= <S',F'>, algebraic signatures, a signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ is a pair $< \sigma_S:S \rightarrow S', \sigma_F >$ where σ_S is a function and $\sigma_F = < \sigma_{Fn}:F^n \rightarrow F'^n >$ is a family of functions, such that if $\sigma_{Fn}(f) = f'$ then rank(f)=us and rank(f')=$\sigma^*_S(u)\sigma_S(s)$.

The function rank returns the rank of a function, or of a set of functions. The notation σ_{Fus} refers to the restriction of σ_{Fn} whose domain is the set of functions with rank us. In order to have a morphism between two signatures it is necessary to have a function σ_{Fn} for each rank length n from the domain signature. Thus, for each us from the domain signature, there must have at least one operation name in the rank $\sigma^*_S(u)\sigma_S(s)$.

Example: There is not a morphism between the two signatures Σ_1= <{a,b,c}, <{F_{abc}, F_{aaa}}>> and Σ_2 = <{p,r}, <{F_{ppr}}, {F_{pr}}>>. From Σ_1 to Σ_2, an attempt to define $\sigma_{\{a,b,c\}}$ could be linking a and b to p and c to r. In this way, it would be possible to define σ_{Fabc}:$F_{abc} \rightarrow F_{ppq}$. But σ_{Faaa}:$F_{aaa} \rightarrow F_{ppp}$ would remain undefined as F_{ppp} does not exist. In the opposite side, the option would be linking both p and r to a so that we could define σ_{Fppr}:$F_{ppr} \rightarrow F_{aaa}$. But, again, the rank pr would not have a corresponding aa in Σ_1.

Two signature morphisms can be composed in an associative way if the codomain of the first signature morphism is the domain of the second. By definition, for each signature Σ there exists a special morphism $\sigma_{Id\Sigma}$ linking the signature to itself. It is named identity morphism, and have the following property: $\sigma \circ \sigma_{Id\Sigma} = \sigma = \sigma_{Id\Sigma} \circ \sigma$, for σ: $\Sigma \rightarrow \Sigma'$. The collection of algebraic signatures and signature morphisms with identities and the associative composition operation form a category which will be called *Sign*. Algebraic signatures are the *objects* of the category and signature morphisms are the *morphisms* of the category.

2 Categorical Limit

Limit is a categorical construction that is performed in any categorical diagram (any commutative set of objects linked by morphisms). It gives the more informative object that can be mapped to any of the objects of the diagram. In this way, the limit embodies the semantic intersection of the objects of the diagram. We call a *cone* (Figure 1, left) a diagram with morphisms f_i: o $\rightarrow$ o_i such that for any morphism g:$o_i \rightarrow o_j$, g ° f_i = f_j. We use the notation {f_i: o $\rightarrow$ o_i} for such a cone. In the following we define *limit* for a diagram (Figure 1, right).

Definition 3. A limit for a diagram D with objects o_i is a cone {f_i: o $\rightarrow$ o_i} such that for any other cone {f'_i: o' $\rightarrow$ o_i}, for D, there is a unique morphism !:o' $\rightarrow$ o for which f_i ° ! = f'_i, with f'_i: o' $\rightarrow$ o_i.

$$\begin{tikzcd}
o_i \arrow[rr, "g"] & & o_j \\
& o \arrow[ul, "f_i"] \arrow[ur, "f_j"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
o' \arrow[rr, "!"] \arrow[dr, "f'_i"'] & & o \arrow[dl, "f_i"] \\
& o_i &
\end{tikzcd}$$

Figure 1. A Cone and a Limit for a diagram with object o_i and morphism g:$o_i \rightarrow o_j$

The requirement "for any other cone {f'_i: o' $\rightarrow$ o_i}, for D, there is a unique morphism !:o' $\rightarrow$ o for which ! ° f_i = f'_i, with f'_i: o' $\rightarrow$ o_i" ensures that any other object that could also play the role of vertex of the cone, can be mapped in the limit. In other words, the limit is the more informative object in the category that can be mapped by the morphisms f_i to any object of the diagram. The limit of a digram $o_1 \leftarrow o \rightarrow o_1$ is named *equalizer*. Each morphism gives a "view" of o into o_1. The equalizer expresses the similarities between both views, that is, the semantic intersection of the two maps o

$\rightarrow o_1$. We point the interested reader to [6,12] for complete references on Category Theory.

2.1. Limit in the Category of Algebraic Signatures

A limit in the category of algebraic signature is a pairwise limit: the limit of sorts and the limit in the family of functions.

Definition 4. Let $\Sigma = <S,F>$ and $\Sigma_i = <S_i,F_i>$. Given a diagram D with objects Σ_i and morphisms φ φ_j, a D-cone $\rho_i:\Sigma \rightarrow \Sigma_i$ is a limit of D if and only if, $\rho_{iS}:S \rightarrow S_i$ is the limit of sorts, and,F is the family of limits $\rho_{iFn}:F^n \rightarrow F^n{}_i$ for each n, natural number.

Following, we exemplify limit in the category of Algebraic Signatures considering a particular case named *product*: a limit performed in a diagram composed by any two objects, without morphisms linking them.

Example: Consider the following signatures: $\Sigma_1 = < \{a,b\},F_1{}^2, F_1{}^3>$, where $F_1{}^2 = \{f:a \rightarrow b\}$, $F_1{}^3 = \{f:a \times a \rightarrow b\}$. $\Sigma_2 = < \{p\},F_2{}^2>$, where $F_2{}^2= \{f:p \rightarrow p, g:p \rightarrow p\}$. The product $\Sigma_1 \times \Sigma_2$ is the signature $\Sigma_{1\ \times 2}= < \{ap,bp\},F_{1\times 2}{}^2>$, where $F_{1\times 2}{}^2 = \{ff:ap \rightarrow ap, fg:ap \rightarrow ap\}$. The first projection is π_1: $\Sigma_{1\times 2} \rightarrow \Sigma_1 = < \pi_{1S},\pi_{1F1\times 2} >$, where π_{1S} maps ap to a and bp to b ; $\pi_{\ 1F1\times 2}$ maps $ff:ap \rightarrow ap$ to $f:a \rightarrow a$ and $fg:ap \rightarrow ap$ to $f:a \rightarrow a$. The second projection is similar.

2.2. The Category of Algebraic Specifications

We define the category of specifications, where objects are pairs composed by signature and set of equations. We define limit in this category, which we will call *Spec*.

An algebraic specification is a pair $Sp = <\Sigma,\Phi>$ where Σ is an algebraic signature and Φ is a set of equations constructed with the symbols of Σ, called Σ-equations.

Definition 5. Given $Sp = <\Sigma,\Phi>$ and $Sp' = <\Sigma',\Phi'>$, algebraic specification, an specification morphism from Sp to Sp' is a pair $<\sigma:\Sigma \rightarrow \Sigma', \alpha >$ such that σ is a signature morphism and $\alpha:\Phi \rightarrow \Phi'$ exists if and only if for all $e \in \Phi$, $\Phi' \vdash Sen(\sigma)(e)$.

In the above definition, Sen: *Sign* → *Set* maps a signature to the set of all sentences that can be written in that signature. Sentences (equations) are pairs of terms of the same sort denoted by $t_i = t_j$, $i,j < \omega$. Terms of a signature are those obtained by the application of the two following items, for a given signature $<S,F>$, and a set Xs of variables for each $s \in S$: (i) A variable $x \in Xs$ is a term of sort s ; (ii) If $t_1,..., t_n$ are term of sorts $s_1,..., s_n$, and $f: s_1,..., s_n \rightarrow s$ is an operation symbol of the signature then $f(t_1,..., t_n)$ is a term of sort s. $Sen(\sigma)$: $Sen(\Sigma) \rightarrow Sen(\Sigma')$ is a function that translates sentences of Σ to sentences of Σ'. $Sen(\sigma)(e)$ denotes the translation of an equation e of Φ to the signature Σ'. The collection of algebraic specifications and specifications morphisms form a category which will be called *Spec*.

Theorem 1. Let $Sp = <\Sigma,\Phi>$ and $Sp_i = <\Sigma_i,\Phi_i>$. Given a diagram D with objects Sp_i and morphisms $<\varphi_j,\alpha_j>$, a D -cone $<\rho_i,\beta_i:\Phi \rightarrow \Phi_i >:Sp \rightarrow Sp_i$ is a limit of D if and only if, ρ_i is the limit of signatures, and, $\Phi = \cap\Psi_i$, where $\Psi_i = \{e \mid Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$.

Proof: We The proof that ρ_i is the limit of signatures can be found in [7]. We present here the component of equations. By definition, Φ contains all equations that, when properly translated, belong to the consequences of each Φ_i, and nothing more than this. Supose that Φ' is another set of equations that could be the component of equations in the limit. Then, there is a cone $\Phi' \rightarrow \Phi_i$. Hence, any equation of Φ', with the proper translation, is in $Cn(\Phi_i)$, for all Φ_i . Consider the three options: (i) Φ' has not all the equations that, when translated, are in $\cap Cn(\Phi_i)$. Then, there is a morphism $\Phi' \rightarrow \Phi$. (ii) Φ' has all, and nothing more than the equations that, when translated, are in $\cap Cn(\Phi_i)$. Then, there is an isomorphism $\Phi' \rightarrow \Phi$. (iii) Φ' has more than the equations that, when translated, are in $\cap Cn(\Phi_i)$. Then, the cone $\Phi' \rightarrow \Phi_i$ do not exist, and Φ' is not a candidate of limit. It is sufficient to show the existence of the morphism $\Phi' \rightarrow \Phi$ because unicity is ensured by the definition of morphisms in *Spec*.

3 Example of limit

In [7] we presented a case study where a pre-order specification was derived from specifications of partial order and equivalence relation. A partial order has the properties of reflexivity, antisymmetry and transitivity. An equivalence relation has the properties reflexivity, symmetry and transitivity. The limit of the diagram composed by both specifications (without morphisms) gives the semantic intersection of them: a reflexive and transitive specification. In this paper we present specifications of Heyting Algebra (HA) and Boolean Algebra (BA). We show that the limit of these specifications is a Bounded Lattice. Next, we use equalizers to hide a property of the Bounded Lattice, creating a specification of Lattice.

Boolean algebra is a structure $<B,\wedge, \vee, ',0,1>$, where B is a set, $\wedge,\vee$ are binary operations, ' is an unary operation and 0,1 are nullary operations. In the specifications of Figure 2 the first eight equations characterizes the substructure $<B,\wedge,\vee >$ as a lattice. The nineth equation (which implies in the dual $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$) adds distributive property to the previous lattice. Extending this substructure with 0 and 1, and considering the tenth and eleventh equations, the resulting object is named a bounded lattice. Finally the last two equations give meaning to the operation ', and completes the definition of a Boolean Algebra.

```
BA=
Sort B
Opns  ∧,∨: B  × B → B
          ': B → B
        0,1: B
∀x,y,z: B
1. x ∧ y = y ∧ x
2. x ∨ y = y ∨ x
3. x ∧ (y ∧ z) = (x ∧ y) ∧ z
4.  x ∨ (y ∨ z) = (x ∨ y) ∨ z
5.  x ∧ x = x
6.  x ∨ x = x
7.  x = x ∧ (x ∨ y)
8.  x = x ∨ (x ∧ y)
9. x ∧ (y ∨ z) = (x ∧ y) ∨ (x ∧ z)
10.  x ∧ 0 = 0
11.  x ∨ 1 = 1
12.  x ∧ x' = 0
13.  x ∨ x' = 1
```

Figure 2. Specification of Boolean Algebra (BA)

Heyting algebra is a structure $\langle H,\wedge,\vee,\rightarrow,0,1\rangle$, where H is a set, $\wedge,\vee,\rightarrow$ are binary operations and 0,1 are nullary operations. As in BA, the first eight equations characterizes the substructure $\langle H,\wedge,\vee\rangle$ as a lattice. The nineth equation is, again, to add distributive property to the lattice. Extending this substructure with 0 and 1, and considering the tenth and eleventh equations, we have a bounded lattice. The last five equations complete the definition of a Heyting Algebra. The specification of Heyting Algebra is shown in Figure 3.

```
HA=
Sort H
Opns  ∧,∨,→: H × H → H
      0,1: H
∀x,y,z: H
1. x ∧ y = y ∧ x
2. x ∨ y = y ∨ x
3. x ∧ (y ∧ z) = (x ∧ y) ∧ z
4. x ∨ (y ∨ z) = (x ∨ y) ∨ z
5. x ∧ x = x
6. x ∨ x = x
7.  x = x ∧ (x ∨ y)
8.  x = x ∨ (x ∧ y)
9. x ∧ (y ∨ z) = (x ∧ y) ∨ (x ∧ z)
10. x ∧ 0 = 0
11. x ∨ 1 = 1
12. x → x = 1
13. (x → y) ∧ y = y
14. x ∧ (x → y) = x ∧ y
15. x → (y ∧ z) = (x → y) ∧ (x → z)
16. (x ∨ y) → z = (x → z) ∧ (y → z)
```

Figure 3. Specification of Heyting Algebra (HA)

In *Sign*, the limit of BA and HA results in the signature $\langle BH, \wedge\wedge, \wedge\vee, \wedge\rightarrow, \vee\wedge, \vee\vee, \vee\rightarrow, 00, 01, 10, 11\rangle$. The limit of singletons sets of sorts results in the unique singleton {BH}. The limit of the set of two binary operations by the set of three binary operations results in the set of six elements $\{\wedge\wedge, \wedge\vee, \wedge\rightarrow, \vee\wedge, \vee\vee, \vee\rightarrow\}$. Finally, the limit of the two sets of nullary operations results in the set of four elements {00, 01, 10, 11}. Thus, the operation ' is out of the limit. In *Spec*, the limit of BA and HA is the specification with signature $\underline{BA \times HA}$ = {Sort BH, Opns $\wedge\wedge, \wedge\vee, \wedge\rightarrow, \vee\wedge, \vee\vee, \vee\rightarrow$: BH×BH → BH, 00,01,10,11: BH}. It results a specification with unnecessary (redundant) symbols. For example, the presence of 01, 10 among the unary operations is not necessary to the semantics of the product BA × HA as a bounded lattice: these symbols do not add any meaning, and their abcense do not loose any meaning. We will show in section 4 how to use equalizers to eliminate undesirable symbols.

The set of equations of the product is not extensively generated, but well defined, according to Theorem 1. It is easy to see that equations of a distributive lattice, and equations that give meaning to 0 and 1 belong to sets of consequences of both BA and HA, and thus, will be present in the product specification. These are those equations written with duplicated symbols, like $x \wedge\wedge 00 = 00$. In this equation, x is a variable of sort BH, and operations ∧∧ and 00 are binary and nullary operations on this same sort. Other equations that will also be present in the product will be some of those where the symbols are not duplicated, like in x ∧→ 01 = 01. This kind of equation specifies a parallel behavior that do not invalidate the wanted meaning, but we are not interested in. Thus, they may be eliminated. The limit specification is shown in Figure 4.

```
BA×HA=
Sort BH
Opns  ∧∧,∧∨,∧→: BH × BH → BH
      ∨∧,∨∨,∨→: BH × BH → BH
      00,01,10,11: BH
∀x,y,z: BH
{e | Sen(σ1)(e) ∈ Cn(BA)} ∩ {e | Sen(σ2)(e) ∈ Cn(HA)}
```

Figure 4. Limit Specification of Boolean Algebra (BA) and Heyting Algebra (HA)

4 The Equalizer

As the set of equations may not be given extensively, to eliminate symbols of specification it is necessary to define an operation to perform changes in this set, ensuring the meaning. We use equalizers with this purpose because the definition of limit ensures that every equation of the limit specification can be proved in the specifications of the diagram by the existence of morphisms from the limit to any specification of the diagram, thus, when deleting symbols (signature components), the exclusion of equations happens as a consequence as equalisation is performed in the category of specification.

Equalizer is the limit in a diagram formed by two objects linked by to morphisms. To use equalizers to omit symbols of a specification, we must construct the diagram as we describe in the following. The specification in the domain of the morphisms is the original specification ($\underline{BA \times HA}$). The codomain is equal to the domain, except that, for each name to be excluded, it has an additional unused name (with the same rank size, in case of operations). Let us call the auxiliary specification of our example $\underline{BA \times HA}^{+}$. It's signature is shown in Figure 5, at right.

```
BA×HA+=
Sort BH,ZERO
Opns  ∧∧,∧∨,∧→: BH × BH → BH
      ∨∧,∨∨,∨→: BH × BH → BH
           THREE:ZERO × ZERO → ZERO
   00,01,10,11: BH
             ONE: ZERO

LATICCE=
Sort BH
Opns  ∧∧,∨∨: BH × BH → BH
      00,11: BH
```

Figure 5. Extended signature to perform the equalizer and the resulting Laticce

We use the names ZERO, THREE and ONE in allusion to the rank size of the symbols to be excluded. The set of equations of this specification is the same as the original one. The morphisms of the equalizer are an inclusion (i: $BA \times HA \rightarrow BA \times HA^{+}$) and a morphism (e:$BA \times HA \rightarrow BA \times HA^{+}$) equal to the inclusion, except that it links the symbols we want to omit to the new symbols of the same rank size. There is no doubt that the inclusion is a morphism in *Spec*. As we use the same set of equations, we ensure that e is also a morphism in *Spec*. The result signature is *LATTICE*, shown in Figure 5, at left.

By definition of equalizer, it has all the symbols of *BA × HA* that are linked to the same symbols by both i and e. The set of equations is not extensively listed, but by definition of limit, it has all equations that can be written in the reduced signature and belong to both sets of consequences.

To finalize this section, we must justify the option for equalizing in the category *Spec* while some approaches just hide symbols of signatures [14].

If we consider the usual semantics of hiding symbols at model level, we see that the meaning of a specification that had an omitted symbol is not always the same meaning of a specification who was first constructed without that symbol. This happens because the semantics of the operation to hide symbols is based on reduct algebras [15]. Considering, now, semantics at theory level, we see, by the following example, that something similar happens. If we omit symbols in an specification (together with the set of equations where they appear) and get the semantics, the wanted meaning is not the same as an specification who was first constructed without these symbols.

Consider the equation 0 = 1', which is in the set of consequences of *BA* (the number superscript before the equality symbol refers to *BA* equation numbering in Figure 2):

$0 \,^{12}{=}\, 1 \wedge 1' \,^{6}{=}\, 1 \wedge (1' \vee 1') \,^{13}{=}\, (1' \vee 1) \wedge (1' \vee 1') \,^{9}{=}\, 1' \vee (1 \wedge 1') \,^{12}{=}\, 1' \vee 0 \,^{10}{=}\, 1' \vee (1' \wedge 0) \,^{8}{=}\, 1'$.

Note that, in the set of equations of *BA*, all equations make use of symbols $\wedge$ and $\vee$. If we derive a new specification by omitting in *BA* symbols $\wedge$ and $\vee$, what would happen in the resulting theory? To answer this question just considering the set of equations in *BA* we would say that the new specification would have an empty set of equations. Clearly, this is not what we would expect since the equation 0 = 1', in which the omitted symbols do not appear, would not be in the set of consequences of the resulting specification. What we would expect is a set of equations whose consequences include all equations from the previous set of consequences, where the $\wedge$ and $\vee$ do not appear.

Using equalizers to obtain the new specification we ensure the presence of everything that can be written with the remaining symbols and that is a member of the original set of consequences. The diagram to be equalised would be composed by specification *BA* and *BA*$^{+}$, where the latter differs form the former just in an additional binary operator. The morphisms would be the inclusion, and the morphism e, equal to the inclusion except in linking $\wedge$ and $\vee$ to the additional operation. The morphisms are shown in Figure 6.

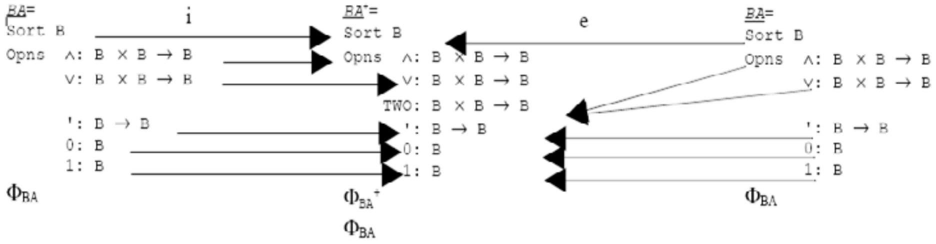


Figure 6. Inclusion signature morphisms i in (BA) and Exclusion signature morphisms e in (BA)

The resulting signature would be <u>*NEWSPEC*</u> = {Sort B, Opns ' : B → H, 0, 1 : B}, with set of equations $\Phi = \{e \mid \text{Sen}\ \sigma\ (e) \in \text{Cn}(\Phi_{BA})\}$. That is, Sen σ(e) translates any equation written with 0,1 and ' to the signature of BA. An equation e will be in the new set of equations if it belongs to the consequences of <u>*BA*</u>. Thus, 0=1' is in the set of equations of <u>*NEWSPEC*</u>.

5 Related and Further Works

This paper extends [7], where limit is proposed as a way of formalizing the semantic intersection of specifications. In this paper we extend the applicability of limit, proposing the use of equalizers (a particular case of limit) as one more categorical concept in the definition of operations on algebraic specifications. We show that the concept is easy to be instantiated in the category *Spec* of algebraic specifications, and also that it has interesting applications making possible to hide components of specifications, reducing its functionality.

[13] also considers the problem of hiding symbols of specifications, but defines operation (*hide* or *derive*) using immersions in *Sign*. In this approach, the effect in the set of equations must be defined separately. In the approach we present here the effect in equations is a consequence of equalizing in *Spec*.

The concept of limit of specifications can also be found in [16]. Indexed categories are used, what makes the approach complicated and different from the usual jargon. [11] constructs the pullback (a particular case of limit) in some categories to formalize specifications, differing in the kind of morphism used. But some restrictions must be considered in order to state the concept of limit.

Recent researches are being developed in the direction of bringing the benefits of the use of categorical constructs to other fields [1]. As example, we can cite the Semantic Web, where integration and interoperability of heterogeneous entities are key concepts. In this case, the concept of limit can be largely explored, as the employed categories do not present the intricacies of *Spec*. Among the categorical approaches that considers limit in this field we can cite [5], who suggests the use of limit (product and pullback) to formalize intersection and translation refinement of ontologies, and [17] uses limit to formalize intersection of alignments of ontologies.

The results presented in this paper are part of a project that investigates the construction of a general algebraic framework to compose/decompose entities in different domains of applications. The power of abstraction of Category Theory is explored to capture common properties of different applications and state the minimum requirements that such a framework must have. As colimit has been largely used to formalize composition, we argue, by dualization, that limit should also be explored to provide rigorous basis to formalize decomposition of entities. We consider limitations and intricacies of some fields of application that makes difficult the definition of limit and investigate how categories should be defined in order to simplify this definition. In this sense, we adopt in this paper, an alternative definition of algebraic signatures to complete the algebraic framework to compose/decompose specifications. In the same direction, in [9], we proposed a categorical formalization of ontologies in order to define an algebraic framework to compose/decompose ontologies aiming the semantical interoperability in the semantic web.

6 Conclusion

Reverse Engineering on libraries of specifications requires a framework in which specifications can be manipulated. Many frameworks propose operations that are traditionally used to combine specifications [10,13,15,16]. In all cases, colimit is the categorical concept that, in an elegant and uniform way, characterizes these modular constructions.

Contrasting to this, operations to break specifications are not easily found, and usually do not present simple and elegant formalization. This fact is relevant because Category Theory offers the concept of limit that turns out to be the dual operation of colimit. As colimit formalizes operations to put things together, limit should formalize operations to break things. But in the Category of Algebraic Specifications, the definition of limit is not as easy as colimit.

In [7] we discussed this situation and proposed an alternative formalization of Algebraic Specifications, very closer to the usual one, where both limit and colimit can be easily defined, and thus, specifications can be composed or broken.

In this paper we complete this approach proposing equalizers (a particular case of limit) to formalize operations that make possible to extract parts of specification. We also present an example that illustrates how (general) limit and equalizers work. With these definitions we complete an algebraic framework to manipulate specifications enabling both the composition and decomposition of existing specifications. This framework is well founded in Category Theory, what ensures that implementation can be done over a solid foundation, avoiding errors and inconsistencies.

References

[1] T. Bench-Capon and G. Malcolm,Formalizing Ontologies and their Relations, in: Kim Viborg Andersen, John K. Debenham, Roland Wagner, eds., *Proceedings of 16th International Conference on Database and Expert Systems Applications (DEXA'99)* (1999).

[2] Braga, Rosana T. V. Engenharia Reversa e Reengenharia, http://www.inf.ufpr.br/silvia/ES/reengenharia

[3] Chikofsky, Cross and May, Reverse engineering, in Advances in Computers, Volume 35

[4] The Common Framework Initiative for algebraic specification and development http://www.informatik.uni-bremen.de/cofi/

[5] J. Jannink, S. Pichai, D. Verheijen, and G. Wiederhold, Encapsulation and Composition of Ontologies, *Proceedings of the AAAI98* (1998).

[6] R. Goldblatt, *Topoi: The Categorical Analysis of Logic*, ser. Studies in Logic and the Foundations of Mathematics, **98** (NorthHolland, 1979)

[7] I. Cafezeiro and E. H. Haeusler, Categorical Limits and Reuse of Algebraic Specifications, in *Advances in Logic, Artificial Intelligence and Robotics*, Eds J. MihoroAbe and J. Silva, IOS Press, Amsterdan, (2002) 216--233.

[8] I. Cafezeiro and E. H. Haeusler, Limits and other Categorical Concepts in Algebraic Signatures, RT-07/02, Technical Reports, www.ic.uff.br/PosGrad/reltec.html, Universidade Federal Fluminense, Niteroi, RJ, Brasil (2003).

[9] I. Cafezeiro and E. H. Haeusler, Semantic Interoperability via Category Theory, Conferences in Research and Practice in Information Technology, Vol. 83. J. Grundy, S. Hartmann, A. H. F. Laender, L. Maciaszek and J. F. Roddick, Eds. 26th International Conference on Conceptual Modeling, ER 2007, Auckland, New Zealand.

[10] H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specifications*. Equations and Initial Semantics, Springer-Verlag, Berlin (1985).

[11] H. Ehrig and F. Parisi-Presicce, Nonequivalence of categories for equational algebraic specifications, in: M.Bidoit, C.Choppy, eds., *Recent Trends in Data Type Specification 8th ADT/ 3rd COMPASS Workshop*, Lect.Notes Comp.Sci. **655** (Springer-Verlag, 1993) 222-235.

[12] S. MacLane, *Categories for the Working Matematician*, Berlin: Springer Verlag(1997).

[13] F. Orejas, Structuring and Modularity, in *Algebraic Foundations of Systems Specification*, Eds E. Artesiano, H. Kreowski, and B. Krieg-Bruckner, Springer, Berlin (1999) 159 - 200.

[14] A. Tarlecki, J. Goguen and R. M. Burstall, Some fundamental algebraic tools for the semantics of computation. Part III: Indexed categories. Theoretical Computer Science **91** (1991).

[15] A. Tarlecki,D. Sannella, Algebraic Preliminaries in Algebraic Foundations of Systems Specification, Eds E. Artesiano, H. Kreowski, and B. Krieg-Bruckner, Springer, Berlin (1999) 13 - 30.

[16] R. Waldinger, Y.V. Srinivas, A. Goldberg, and R. Jullig, Specware Language Manual, Suresoft,Inc, Kestrel (1996).

[17] A. Zimmermann, M. Krotzsch, J. Euzenat and P. Hitzler,Formalizing Ontology Alignment and its Operations with Category Theory, *Proceedings of the 4th International conference on Formal ontology in information systems (FOIS'06)*(Baltimore, Maryland,2006).

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-13

An Attempt to Express the Semantics of the Adaptive Devices

Ricardo Luis de Azevedo da ROCHA
Computing Engineering Department, Escola Politécnica da USP, Brazil
and
Computing Engineering, Faculdade de Engenharia da Fundação Santo André, Brazil

Abstract. Adaptive devices were introduced formally in [7] as a generalization of the adaptive automaton [6]. The semantics were defined by an entirely traditional operational way, based on automaton transitions. This paper starts another path, to describe the semantics of adaptive devices using structural operational semantics [9, 12], λ-calculus [1, 2] and a typing system based on the algorithm **W** of Damas and Milner [3, 8]. The results achieved showed that adaptive devices are complex to describe, and that a sound and complete typing system should be used to fully analyze them.

Keywords. structural operational semantics, adaptive systems, λ-calculus, untyped systems.

Introduction

The devices that are considered in this paper are the rule-driven devices. Those devices are static; their internal structure remains unchanged over any computational task. Their behavior is mapped by a relation among their own internal status (state, rule, etc) and an input stimulus to another internal status and a possibly empty output event. In this sort of device one can view automata, grammars, logic, programs, etc.

Let us call them ordinary devices. They are ordinary in the sense that they cannot change their own internal structure without any external intervention. This way, their structure must be defined before their use and remain unchanged, until some designer edits the structures and creates new devices.

The adaptive technology intends to orderly break this process (the edit-start-stop cycle) by endowing any ordinary device with the ability to change its own internal structure without the need of external intervention (editing). But the changes must follow some rules, and happen according to some predefined constraints, in other words following some '*changing program*'.

An adaptive device is a computational device composed by two parts, an ordinary device (any finite rule-based device, transition system, rewriting-system) encapsulated by an intrusive mechanism that has the ability to change the encapsulated ordinary device. Although the encapsulating structure does not interfere in the semantic actions performed by the ordinary device, it does indeed intervene in the way that these actions are triggered.

By means of this interference the encapsulated ordinary device becomes enabled to change its own structure (in an automaton its topology), and, using it's modified (new) structure, becomes also enabled to deal with some new features, to which it was not originally prescribed. The main goal of adaptive models is to use simpler models to build complex structures, such as in [6]. This research use an automaton-based model and through some features acquire it with the ability to perform much more complex actions - for example, in order to accept strings from recursively enumerable (type 0) languages [6].

A device is considered adaptive whenever its behavior changes dynamically in a direct response to its input stimuli, without interference of external agents [7]. To achieve this feature adaptive devices have to be self-modifiable. In [7] this definition is introduced in order to extend it to any rule-driven device.

1. Adaptive Devices

Let an ordinary device be a *k*-tuple M_i, which has a computing function $d(M_i)$, also let $\boldsymbol{M}$ represent the space of ordinary devices ($M_i \in \boldsymbol{M}$). Any environmental input to the device (for example, an input string for an automaton, or a sentential form for a rewriting-system) is not being considered for now. The device represents one model over the whole space of ordinary devices of the same kind.

General adaptive devices were introduced formally in [7] as a generalization of the adaptive automaton [6]. The semantics was defined by an entirely traditional operational way, based on automata transitions, (even when the device is not transition-based). The definition of the adaptive actions was done in an informal level, describing the activities in natural language, thus creating different interpretations from the same definition.

In order to deal with these problems it is necessary to formalize the definitions. And it seems to be a natural choice to use the Structural Operational Semantics approach. But to formally define the structural operational semantics, following [9], of the adaptive devices it is necessary first to informally describe what is meant by an adaptive action.

An adaptive action is defined as a function call in which parameters are passed in a call-by-value strategy. Any adaptive action potentially has local variables that are filled once by elementary inspection actions (defined below) and generators that are filled once with new values (different from all values or variables defined in the model) when the adaptive action starts. The execution of an adaptive action follows the execution of a function that is called by a computing agent. But as there may be structural changes, and also insertion of structural computing elements (with entirely new values), there is an execution order for an adaptive action.

Definition 1 *[Order of Execution] The adaptive action is performed by the sequence below [6, 7]:*

1. *perform an adaptive action (before anything happens), the only values available for this action are those that came from parameters;*
2. *fill the generators;*
3. *perform the set of elementary adaptive actions of inspection (in parallel);*
4. *perform the set of elementary adaptive actions of deletion (in parallel);*
5. *perform the set of elementary adaptive actions of insertion (in parallel);*
6. *perform an adaptive action (after all the other actions have happened) all the values are available for this action.*

The adaptive actions called inside an adaptive action happen to be performed and defined in the same way. So all the values are only available after the elementary inspection actions have been performed, after that the elementary deletion actions can happen, and only after them the elementary insertion actions can happen, then afterwards, the adaptive action to be performed after all the other actions is called, finishing the execution.

Definition 2 *[Elementary adaptive actions] Let* $\langle rule \rangle$ *be the pattern of a rule defined for the ordinary encapsulated device (*$\langle rule \rangle \in \delta(M_i)$*), then the elementary adaptive actions are:*

- *Inspection actions, represented by* $?[\langle rule \rangle]$*, which return all the rules of the computing function matching the pattern* $\langle rule \rangle$*.*
- *Deletion actions, represented by* $-[\langle rule \rangle]$*, which delete all the rules of the computing function matching the pattern* $\langle rule \rangle$*.*
- *Insertion actions, represented by* $+[\langle rule \rangle]$*, which inserts the pattern* $\langle rule \rangle$ *to the set of rules of the computing function of the device.*

An adaptive device (*A*-device) is defined as:

Definition 3 *[A-device] An A-device* $A=(M_i, AM_i)$ *is a computational device composed by two parts* M_i *and* AM_i*, or* $A = (M_i, \gamma, \Gamma, \phi, \Phi, \xi, \Xi, \zeta)$*, where* M_i *represents the original encapsulated device underneath the adaptive mechanism,* Φ *is an initially finite set of adaptive actions (containing the null adaptive action* nop*), and* $\delta(M_j)$ *represents the computing function of the device* $M_j \in \boldsymbol{M}$*.*
The set Ξ *is the set of names (variables and values) of the ordinary device. The function* $\xi : \Xi \to \Xi$ *embeds new names into the set* Ξ*. The function* $\zeta : \Xi \times \delta(M_j) \to \delta(M_k)$*, generates new rules for the ordinary device, where* $\delta(M_k)$ *represents a subset of the computing function of the device* M_j *having its rules changed through the substitution of names from set* Ξ*. The partial function* $\Gamma : \delta(M_j) \to \Phi \times 2^{\{?,-\}\times\delta(M_j)\cup\{+\}\times\zeta(\xi(\Xi),\delta(M_j))} \times \Phi$*, generates a, possibly empty, adaptive action. The partial function* $\phi : \Gamma(\delta(M_j) \to \Phi$*, embeds into the set* Φ *an adaptive action. The function* $\gamma : \delta(M_i) \to \Phi \times \delta(M_i) \times \Phi$ *applies to each transition of the A-device* M_i *a, possibly empty, pair of adaptive actions one of them to be performed before the computation step takes place, and the other after the computation step took place.*
An adaptive device is said to be already programmed if the set Φ *is filled before its operation (the execution of the device), also the functions* ϕ *and* Γ *are never used by any adaptive action (in such case the set* Φ *remains unchanged and finite), and the function* γ *is used only to the new rules generated by function* ζ*.*

A general *already programmed A*-device defined above is exactly what was proposed in [6, 7]. The devices of this kind are Turing-powerful as was shown in [11].

2. Abstract Syntax

An adaptive action is, of course, responsible for any change imposed to an A-device. Consider an *already programmed* A-device. For this kind of adaptive device an adaptive action can be described in BNF-like style as:

- $\langle Action \rangle \rightarrow \langle name \rangle(\langle parms \rangle)\{\langle VarGenList \rangle \langle body \rangle\}$
- $\langle VarGenList \rangle \rightarrow \langle GenList \rangle \langle VarList \rangle \mid nil$; where *nil* means empty list, $\langle GenList \rangle$ may be empty, and $\langle VarList \rangle$ may also be empty.
- $\langle body \rangle \rightarrow \langle BeforeAction \rangle \langle ElementaryActionList \rangle \langle AfterAction \rangle$
- $\langle BeforeAction \rangle \rightarrow \langle CallAction \rangle$
- $\langle AfterAction \rangle \rightarrow \langle CallAction \rangle$
- $\langle ElementaryAction \rangle \rightarrow \otimes[\langle rule \rangle]$; where $\otimes \in \{+,-,?\}$

The execution of insertion and deletion elementary adaptive actions may depend directly on the results of the inspection elementary adaptive actions. Just because those actions can delete or insert rules from which only a few parts of information are available and they need to be completely identified, so this is done by inspection elementary actions.

Let $\langle rule \rangle = (c_1, c_2, \ldots, c_n)$, where c_i $(1 \leq i \leq n)$ are constant elements, be a rule of the ordinary device. An incompletely specified $\langle rule \rangle$ is a rule in which there is at least one variable element in the place of a constant. Then a substitution of a name in a rule by an element (value of a variable, generator or parameter) is defined as:

Definition 4 *[Substitution] Let $[x \rightarrow s]$ $\langle rule \rangle$ represent the substitution of x by s in $\langle rule \rangle$, and let $\langle rule \rangle = (c_1, c_2, \ldots, c_n)$, an n-tuple of constant elements c_i $(1 \leq i \leq n)$.*

$[x \mapsto s](c_1) = (c_1)$, if $c_1 \neq x$;
$[x \mapsto s](c_1) = (s)$, if $c_1 = x$;
$[x \mapsto s](c_1, c_2, \cdots, c_n) = ([x \mapsto s](c_1), [x \mapsto s](c_2, \cdots, c_n))$;
$[x \mapsto s] \otimes \langle rule \rangle = \otimes [x \mapsto s] \langle rule \rangle$, where $\otimes \in \{+,-,?\}$.

The inspection elementary adaptive actions need to use the values fulfilled by the parameters and the generators, being responsible for the correct filling of the variables. The terms inside an adaptive action are:

Definition 5 *[Elementary Action Terms] Let $\mathcal{G}$ be an enumerable set of generator names, $\mathcal{V}$ be an enumerable set of variable names, $\mathcal{P}$ be an enumerable set of parameters, $\mathcal{K}$ be the enumerable set of constants (the values inside the rules), and* **A** *be the enumerable set of names of adaptive actions. The set of elementary action terms is the smallest set* T_e *such that:*

1. $\otimes \langle rule \rangle \in T_e$, where $\otimes \in \{+,-,?\}$, for every $\langle rule \rangle$ of the ordinary device;
2. if $t_1 \in T_e$, $s \in \mathcal{V}$ and $x \in \mathcal{K}$, then $[x \mapsto s]\ t_1 \in T_e$;
3. if $t_1 \in T_e$, $s \in \mathcal{P}$ and $x \in \mathcal{K}$, then $[x \mapsto s]\ t_1 \in T_e$;
4. if $t_1 \in T_e$, $s \in \mathcal{G}$ and $x \in \mathcal{K}$, then $[x \mapsto s]\ t_1 \in T_e$.

Definition 6 *[List of Elementary Action Terms] The set of list of elementary action terms is the smallest set* $\mathrm{T_e}^{\mathrm{L}}$ *such that:*

1. $\varnothing \subseteq \mathrm{T_e}^{\mathrm{L}}$

2. if $t_1 \in \mathrm{T_e}$, then $t_1 \in \mathrm{T_e}^{\mathrm{L}}$;

3. if $t_1 \in \mathrm{T_e}$ and $t_2 \in \mathrm{T_e}$, then $t_1t_2 \in \mathrm{T_e}^{\mathrm{L}}$;

4. if $t_1 \in \mathrm{T_e}$ and $t_2 \in \mathrm{T_e}$, then $t_2t_1 \in \mathrm{T_e}^{\mathrm{L}}$.

The terms of an adaptive action can now be defined:

Definition *7 [Adaptive Action Terms] The set of action terms is the smallest set* T *such that:*

1. $\varnothing \subseteq \mathrm{T}$

2. if $x_i \in \mathcal{P}(1 \le i \le n), \boldsymbol{B} \in \boldsymbol{A}$, then $\boldsymbol{B}(x_1, x_2, \dots, x_n) \in \mathrm{T}$

3. if $x_i, y_i \in \mathcal{P}(1 \le i \le n), \boldsymbol{B}, \boldsymbol{F} \in \boldsymbol{A}$, then $\boldsymbol{B}(x_1, x_2, \dots, x_n)\ \boldsymbol{F}(x_1, x_2, \dots, x_n) \in \mathrm{T}$;

4. if $t_1 \in \mathrm{T_e}^{\mathrm{L}}$, then $t_1 \in \mathrm{T}$;

5. if $x_i \in \mathcal{P}(1 \le i \le n), \boldsymbol{B} \in \boldsymbol{A}$ and $t_1 \in \mathrm{T_e}^{\mathrm{L}}$ then $\boldsymbol{B}(x_1, x_2, \dots, x_n)\ t_1 \in \mathrm{T}$;

6. if $x_i \in \mathcal{P}(1 \le i \le n), \boldsymbol{F} \in \mathrm{T}$ and $t_1 \in \mathrm{T_e}^{\mathrm{L}}$ then $t_1\ \boldsymbol{F}(x_1, x_2, \dots, x_n) \in \mathrm{T}$;

7. if $x_i, y_i \in \mathcal{P}(1 \le i \le n), \boldsymbol{B}, \boldsymbol{F} \in \boldsymbol{A}$ and $t_1 \in \mathrm{T_e}^{\mathrm{L}}$ then $\boldsymbol{B}(x_1, x_2, \dots, x_n)\ t_1\ \boldsymbol{F}(y_1, y_2, \dots, y_n) \in \mathrm{T}$.

Having the abstract syntax already defined the next step is to construct an axiomatic system to define the semantics of the adaptive actions following [9].

3. Structural Operational Semantics

The operational semantics has to be defined by parts, at least one part for each elementary adaptive action. Consider at first the case of the inspection elementary adaptive action. This action returns a set $\mathcal{M}$ of matching rules $\mathcal{M} = \{\langle rule\rangle\}$(a set of n-tuples), the variables used in the action are filled with the matching values from the set.

The generators are names that are filled with **new values** $g \notin \mathcal{K}$. The values of the generators can be viewed as results of a function based on a Gödel numbering. Consider the Gödel numbering function $gn\colon \mathcal{K}^\infty \to \mathbf{N}$, where $\mathcal{K}^\infty$ means an infinite set whose finite subset is $\mathcal{K}$, a function $m_{gn}\colon \mathcal{K} \to \mathbf{N}$ which gives the greatest Gödel number of the set $\mathcal{K}$ and a function $\aleph\colon \mathbf{N} \to \mathcal{K}$ which embeds a new value to the set $\mathcal{K}$, when applied to $m_{gn}(\mathcal{K})$ it returns the value of $gn^{-1}(m_{gn}(\mathcal{K})+1)$.

3.1. Inspection Elementary Actions

The inspection actions try to substitute free occurrences of variables for constants which could match some of the rules of the ordinary device underneath.

Definition 8 *[Free Variables] The set of free variables of an inspection elementary adaptive action* e_i, *written* $FV(e_i)$, *is defined as follows:*

- $FV(c) = \varnothing, \forall c \notin \boldsymbol{V}$

- $FV(x) = x, \forall x \notin \boldsymbol{V}$
- $FV(\langle rule\rangle) = FV(c_1, c_2, \ldots, c_n) = FV(c_1) \cup FV(c_1, c_2, \ldots, c_n)$

***Definition 9** [Substitution Instances] Let* $t_1 \in \mathrm{T_e}$ *,* $x \in \boldsymbol{V}$ *and, also,* $x \notin FV(t_1)$*, and* $x \in \mathcal{K}$*, then* $\sigma t_1 = \{x, c\}(t_1 \in \mathrm{T_e})$*, if there exists* $\langle rule\rangle \in \delta(M_i)$ *where* $t_1 = [x \mapsto c]\langle rule\rangle$*. This is written*$\sigma t_1 = \{x, c\}$*, and defines a substitution instance from x to c. More generally* $\sigma t_1 = \{(x_1, c_1), (x_2, c_2), \ldots , (x_m, c_m)\}$ *defines the substitution instances from variables* x_i *to* c_i *in term* t_1*.*

***Criterion 10** [Constraint Satisfaction] Let **C** be set of equations* $\{S_i = R_i^{i\in 1,\ldots,n}\}$*, the set **C** is a constraint set. A substitution* σ *unifies an equation* $\boldsymbol{S} = \boldsymbol{R}$ *if the substitution instances* $\sigma\boldsymbol{S}$ *and* $\sigma\boldsymbol{R}$ *are identical.* σ *unifies (or satisfies) **C** if it unifies every equation in **C**.*

There is a problem with this criterion (and some others on definitions above), because there is no distinction between values inside terms. It could be resolved by a typing system, but it is out of the scope of this paper.

Leaving these problems behind, the unification algorithm defined by Robinson [10] gives an answer to the constraint satisfaction problem, a set of tuples of unifiable variables. Each tuple represents the values that the variables may assume to unify the *constraint set*.

The unified resulting set is the constraint satisfaction of all equations together; this means that if a variable is used in more than one equation then the resulting values are the intersection of the values for each equation. Also, if there is more than a single variable in an equation, then all of the variables are unified together (in tuples). Under these assumptions, the resulting operational semantics for the inspection elementary action is defined by the evaluation relation below:

$$\frac{\sigma t_1 = \{(x_i, c_i) \mid c_i \in \mathcal{K}\}}{\{(x_i, c_i) \mid \exists (c_1, c_2, \ldots, c_i, \ldots, c_n \in \{\langle rule\rangle\}\}} \qquad \text{[EINSP]}$$

$$\frac{\dfrac{\sigma t_1 = \{(x_i, c_i) \mid c_i \in \mathcal{K}\}}{\{(x_i, c_i)\}} \wedge \dfrac{\sigma t_1 = \{(x_j, c_j) \mid c_j \in \mathcal{K}\}}{\{(x_j, c_j)\}}}{\{(x_i, c_i), (x_j, c_j) \mid \exists (c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_n \in \{\langle rule\rangle\}\}} \qquad \text{[EINSP1]}$$

$$\frac{\dfrac{\sigma t_1 = \{(x_i, c_i) \mid c_i \in \mathcal{K}\}}{\{(x_i, c_i)\}} \wedge \dfrac{\sigma t_2 = \{(x_i, c_i^{'}) \mid c_i^{'} \in \mathcal{K}\}}{\{(x_i, c_i^{'})\}}}{\{(x_i, c_i)\} \cap \{(x_i, c_i^{'})\}} \qquad \text{[EINSP2]}$$

From these evaluation rules it is immediate that:

Theorem 11 [Soundness] *If* $\{(x_i, c_i), (x_j, c_j)\}$ *is a solution for some constraint set **C** then there exists rules of the ordinary device where and are substituted by* c_i *,* c_j*,* $(c_1 , c_2, \ldots , c_i ,\ldots , c_j , \ldots , c_n) \in \{\langle rule\rangle\}$ *and this is read:* $(c_1 , c_2, \ldots , (c_i)^{xi} , \ldots , (c_j)^{xj} , \ldots , c_n) \in \{\langle rule\rangle\}$*.*

Theorem 12 [Completeness] $\forall(c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_n)$, *there is a constraint set* C *whose solution is the set* $\{(x_i, c_i), (x_j, c_j)\}$.

3.2. Insertion Elementary Actions

The insertion elementary adaptive actions add to the set of rules of the ordinary device some new rules, based on variables (filled by the inspection actions), parameters (filled before execution) and generators (filled at the start of the adaptive action).

Take the results of any inspection elementary adaptive action as a, possibly empty, set of values. The insertion elementary adaptive actions insert rules to the set of rules of the ordinary device. The set of values for the results are specified, in variable values, as *Val*(*x*):

1. $Val(x) = \Phi$
2. $Val(x) = \{(x_i, c_i)\}$
3. $Val(x) = \{(x_i, c_i), (x_j, c_j)\}$

The application of an insertion elementary adaptive action is represented as $t_1 \overset{+}{\to}_{(xi,\,ci)} t_1'$, where t_1 and t_1' represent the set of rules of the ordinary device, and the pairs (x_i, c_i) represent all the values for the variable x_i filled by inspection elementary adaptive actions that are to be used. The operational semantics for the insertion elementary adaptive action is defined by the evaluation relation below:

$$\frac{t_1 \overset{+}{\to}_{\varnothing} t_1'}{t_1' = t_1} \qquad \text{[EINSR]}$$

$$\frac{t_1 \overset{+}{\to}_{\{(x_i,c_i)\}} t_1'}{t_1' = t_1 \cup \{(c_1, c_2, \cdots, c_i, \cdots, c_n) \mid (c_1, c_2, \cdots, \overset{x_i}{(c_i)}, \cdots, c_n) \in \{\langle rule \rangle\}\}} \qquad \text{[EINSR1]}$$

$$\frac{t_1 \overset{+}{\to}_{\{(x_i,c_i)\} \wedge \{(x_j,c_j)\}} t_1'}{t_1' = t_1 \cup \{(c_1, c_2, \cdots, c_i, \cdots, c_j, \ldots, c_n) \mid (c_1, c_2, \cdots, \overset{x_i}{(c_i)}, \cdots, \overset{x_j}{(c_j)}, \cdots, c_n) \in \{\langle rule \rangle\}\}} \qquad \text{[EINSR2]}$$

When the insertion actions are used, from this point of view (without a typing-system), the ordinary device may behave in a non-deterministic fashion (even if it was deterministic before the insertion). And also, some inserted rules may be duplicated or, even worse, useless, because no control is taken for the values received from the inspection actions.

So, the actual size of the device may increase without any increase on its computational power. Consider some finite constant number k representing the increasing rate of the set $\mathcal{K}$ by an adaptive (non-elementary) action, after n computational steps always using this action the device may have grown by a rate of the order $O(k \times n)$. But as the size increase also an increasing number of possible pairs are returned by the use of the inspection elementary adaptive actions, so the rate k may not be constant. In such a case the device may grow by a rate of the order $O(k^n)$.

3.3. Deletion Elementary Actions

Using the results of the inspection elementary adaptive actions, the values filled in the variables, the deletion elementary adaptive actions remove some rules from the ordinary device, those which match the values filled.

The application of a deletion elementary adaptive action is represented as $t_1 \rightarrow^{-}_{(xi,\, ci)} t_1'$, where t_1 and t_1' represent the set of rules of the ordinary device, and the pairs (x_i, c_i) represent all the values for the variable x_i filled by inspection elementary adaptive actions that are to be used. The operational semantics for the insertion elementary adaptive action is defined by the evaluation relation below:

$$\frac{t_1 \overset{-}{\rightarrow}_{\varnothing} t_1'}{t_1' = t_1} \qquad \text{[EDEL]}$$

$$\frac{t_1 \overset{-}{\rightarrow}_{\{(x_i, c_i)\}} t_1'}{t_1' = t_1 / \{(c_1, c_2, \cdots, c_i, \cdots, c_n) \mid (c_1, c_2, \cdots, \overset{x_i}{(c_i)}, \cdots, c_n) \in \{\langle rule \rangle\}\}} \qquad \text{[EDEL1]}$$

$$\frac{t_1 \overset{-}{\rightarrow}_{\{(x_i, c_i)\} \wedge \{(x_j, c_j)\}} t_1'}{t_1' = t_1 / \{(c_1, c_2, \cdots, c_i, \cdots, c_j, \ldots, c_n) \mid (c_1, c_2, \cdots, \overset{x_i}{(c_i)}, \cdots, \overset{x_j}{(c_j)}, \cdots, c_n) \in \{\langle rule \rangle\}\}} \qquad \text{[EDEL2]}$$

When the deletion actions are used, from this point of view (without a typing-system), the ordinary device may behave in a strange fashion (even if it was deterministic before the deletion), because it can stop processing, fail. And also, some deleted rules may turn some parts of the ordinary device useless (unreachable). Again it happens because no control is taken for the values received from the inspection actions.

If a deletion elementary adaptive action removes a rule that is used by the ordinary device to realize its computation, then it behaves abnormally and cannot compute the function as before.

3.4. Adaptive Actions

The semantics for an adaptive action is defined by the elementary action semantics, as mentioned in the introduction. After an adaptive action is performed the ordinary device is changed to another in the space of (*already programmed*) ordinary devices. There are some problems not addressed by the definitions already used such as:

- There is no distinction between values inside terms;
- The ordinary device may behave in a non-deterministic fashion, even if it was deterministic before the insertion;
- The inserted rules may be duplicated;
- The inserted rules may be useless;
- The actual size of the device may increase without any increase on its computational power;
- The actual size may increase unboundedly;
- The ordinary device can stop processing, fail;
- The deleted rules may turn some parts of the ordinary device useless (unreachable).

- The deleted rules can make the device behave abnormally and become unable to compute the function it used to compute before.

4. Structural Operational Semantics using λ-terms

There is another way to formulate a structural operational semantics for adaptive devices, that is using λ-calculus. An example is the definition of the rules as:

$$\langle \boldsymbol{rule^i} \rangle = \lambda z(c_1^{\,i}, c_2^{\,i}, \dots, c_n^{\,i}), \text{ where } c_j^{\,i} \in \mathcal{K}, \text{ are all constants.}$$

Using this definition as a basis the elementary adaptive actions can be defined using λ-terms, and, after that, the adaptive actions can be defined. This approach may be useful and bring less difficulties or problems, when defining the adaptive mechanism.

4.1. λ-Syntax

The syntax of the λ-terms used in this paper follow [1, 2], also the β-reduction and the semantics of the terms follow the same references. Some of the combinators needed for the application to adaptive devices are defined below.

Using [1][ch. 6], a *fixed point combinator* is a term **M** such that $\forall F\ MF = F\ (MF)$, MF is fixed point of F.

Definition 13 *[Combinators]*

Let $\mathbf{Y} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, $\mathbf{Y}$ is a *fixed point combinator.*

Let $\mathbf{I} = \lambda x.x$, $\mathbf{I}$ is the *identity combinator.*

Let $\mathbf{K} = \lambda xy.x$, $\mathbf{K}$.

Let $\mathbf{S} = \lambda xyz.xz(yz)$, $\mathbf{S}$.

Let $\mathbf{T} = \lambda xy.x$, $\mathbf{T}$ is the *True combinator.*

Let $\mathbf{F} = \lambda xy.y$, $\mathbf{F}$ is the *False combinator.*

Definition 14 *[Finite Sequences]*

$[M] \equiv M$,

$[M_0, \cdots, M_{n+1}] \equiv [M_0, [M_1, \cdots, M_{n+1}]]$

Let $\mathbf{CAR}\ y \equiv \lambda x.x\mathbf{T}\ y$, which returns the first element of the sequence y.

Let $\mathbf{CDR}\ y \equiv \lambda x.x\mathrm{F}\ y$, which returns the rest of the sequence y.

Definition 15 *[Numerals] For each $n \in \mathrm{N}$ the term $\ulcorner n \urcorner$ is defined:*

- $\ulcorner \mathbf{0} \urcorner \equiv \mathbf{I}$
- $\ulcorner n+1 \urcorner \equiv [\mathrm{F}, \ulcorner n \urcorner]$

Let $\mathrm{S}^+ \equiv \lambda x.[\mathrm{F}, x], \mathrm{P}^- \equiv \lambda x.x\mathrm{F}, \mathbf{Zero} \equiv \lambda x.x\mathbf{T}$, then:

$$\mathbf{S}^{+}\ulcorner n \urcorner = \ulcorner n+1 \urcorner, \mathbf{P}^{-}\ulcorner n+1 \urcorner = \ulcorner n \urcorner, \mathbf{Zero}\ \ulcorner 0 \urcorner = \mathbf{T}, \mathbf{Zero}\ \ulcorner n+1 \urcorner = \mathbf{F}$$

Let $\mathbf{A\_}\ \boldsymbol{x}\ \boldsymbol{y} \equiv \mathbf{Y}(\mathbf{Zero}\ \boldsymbol{y}\ (\boldsymbol{x})\ (\mathbf{A\_}\ \mathbf{P}^{-}\ \boldsymbol{x}\ \mathbf{P}^{-}\ \boldsymbol{y}))$

Definition 16 *[Gödel Numbers] Let* $\#: \Lambda \rightarrow \mathrm{N}$ *be a bijective mapping,* ***M*** *is called a Gödel number of* ***M****.*

$\ulcorner \# \boldsymbol{M} \urcorner$ defines a numeral for a λ-term M .

Using the definitions 13, 14, 15, 16 above, it is possible to express the semantics of the adaptive actions through the semantics of the λ-terms.

4.2. λ-Definition of Adaptive Terms

The inspection elementary adaptive action may be defined over the β-reductions whose applications could have produced the ordinary device's rules. By this view:

$$(\lambda x_j^i(\lambda z(c_1^i, c_2^i, \cdots, x_j^i, \cdots, c_n^i)))c_j^i \xrightarrow{\beta} \lambda z(c_1^i, c_2^i, \cdots, c_j^i, \cdots, c_n^i),$$

so the algorithm Unify produces the pair (x_j^i , c_j^i).

Definition 17 *[Rule Search] Let* A? *be a combinator that, when applied to a sequence of rules* ***M****, where* ***M*** *has some variables to be replaced by constants, followed by a sequence of rules* ***N*** *produces a resulting sequence* ***M'*** *based on the pairs* [$\boldsymbol{x}_j^i$, $\boldsymbol{c}_j^i$] *where* $\boldsymbol{x}_j^i$ *represents the* $\boldsymbol{j}^{th}$ *variable of the* $\boldsymbol{i}^{th}$ *rule of the sequence* ***N****, and* $\boldsymbol{c}_j^i$ *represents the substitution instance for* $\boldsymbol{x}_j^i$*.*

Lemma 18 *[Inspection Combinator] There is a term* A? *such that* A? ***MN*** *=* ***M'****.*

Proof. The unification algorithm does exactly this job. Based on proposition 6.3.11 from [1], all recursive functions are λ-definable, so there is a combinator A? that performs the Unify algorithm.

From the β-reduction of the values unified all the variables over the adaptive action body become fulfilled after the inspection elementary adaptive actions had performed (see section 4.3 below).

The insertion elementary adaptive action may be defined as a graph inclusion based on de Bruijn nameless terms [1, 2, 4, 5]. The finite sequence of rules form a not necessarily connected graph where the use of a combinator A+ the resulting graph (sequence) is reduced to a bigger one, because its size in number of rules increase.

□

Definition 19 *[Rule Insertion] Let* A+ *be a combinator that, when applied to a rule* ***M*** *followed by a sequence of rules* ***N*** *produces a resulting sequence* [***M***,***N***] *with the rule insert to the original sequence.*

Lemma 20 *[Insertion Combinator] There is a term* A+ *such that* A+ ***MN*** = **[*M*,*N*]**.

Proof. Let **App = λ*xyz.zxy***, then **App *MN*=[*M*,*N*]**, inserting the rule ***M*** in the list.

Let $\mathbf{Tst} = (\lambda\, \boldsymbol{rule}.\, \mathbf{Zero}\, (\mathbf{A}^{-}\ \ulcorner \#\, (\boldsymbol{rule}\ \boldsymbol{c}_j^i) \urcorner\ \ulcorner \#\, (\boldsymbol{rule}\ \boldsymbol{c}_i^i) \urcorner\ ))$, $i \neq j$, then the application of **Tst** on a rule verifies if the rule is in normal-form. If the rule is in normal-form it cannot be reduced. Therefore, the minus operator (**A**$^-$) returns zero ($\ulcorner \mathbf{0} \urcorner$). And if it is not in normal-form it will be reduced to two different values so their Gödel numbers will be different.

Define $\mathrm{A}_+ = (\lambda\, \boldsymbol{x}\ \boldsymbol{y}.\, \mathbf{Tst}\ \boldsymbol{x}\ (\mathbf{App}\ \boldsymbol{x}\ \boldsymbol{y})\ \boldsymbol{y})$. So, let ***M*** be the inserted rule and ***N*** be the finite sequence of rules before the application of A$_+$. Therefore, in this case, the resulting sequence after (A$_+$***MN***) has a greater number of rules than ***N*** if the rule ***M*** is in normal-form.

□

Proposition 21 *[Computability Effect] Any insertion elementary adaptive action preserves the original computational function.*

Proof. Directly from the definition 19 and previous lemma 20, since A$_+$***MN*** = **[*M*,*N*]**, or A$_+$***MN*** = ***N*** and the original rules ***N*** are preserved (they were not removed).

□

The deletion elementary adaptive action may also be defined as a graph deletion based on de Bruijn nameless terms [4]. Using a combinator A$_-$ on the finite sequence of rules yields a resulting sequence which is reduced to a smaller one, because its size in number of rules decrease.

Definition 22 *[Rule Deletion] Let* A$_-$ *be a combinator that, when applied to a rule* ***M*** *followed by a sequence of rules* ***N*** *produces a resulting sequence* ***N'*** *which does not have the rule* ***M*** *in it.*

Lemma 23 *[Deletion Combinator] There is a term* A$_-$ *such that* A$_-$***MN*** = ***N'****, where* $\boldsymbol{M} \notin \boldsymbol{N'}$.

Proof. Let $\mathbf{Eq}\ \boldsymbol{x}\ \boldsymbol{y} = \mathbf{Zero}\, (\mathbf{A}^{-}\ \ulcorner \#\, \boldsymbol{x} \urcorner\ \ulcorner \#\, (\mathbf{T}\ \boldsymbol{y}) \urcorner)$, then **Eq** verifies the equality of the Gödel numbers of the λ-terms $\boldsymbol{x}, \boldsymbol{y}$.

Let

$\mathbf{R} = \mathbf{Y}\ ((\mathbf{Eq}\ \boldsymbol{x}\ \boldsymbol{y})\ (\mathrm{A}_+\ \boldsymbol{z}\ (\mathbf{R}\ \boldsymbol{x}\ (\mathbf{CDR}\ \boldsymbol{y})\ \boldsymbol{z}))\ (\mathrm{A}_+\ (\mathbf{T}\ \boldsymbol{y})\ (\mathbf{R}\ \boldsymbol{x}\ (\mathbf{CDR}\ \boldsymbol{y})\ \boldsymbol{z})))$, then this combinator reconstructs the original sequence $\boldsymbol{y}$ leaving the term $\boldsymbol{x}$ out of it. The term $\boldsymbol{z}$ is used only when there is a match, that is, when the rule has to be removed, so this term must be β-reducible.

Define $\mathrm{A}_-\ \boldsymbol{M}\ \boldsymbol{N} \equiv \mathbf{R}\ \boldsymbol{M}\ \boldsymbol{N}\ \mathbf{I}$. So, let ***M*** be the deleted rule and ***N*** be the finite sequence of rules before the application of A$_-$ then the resulting sequence ***N'*** after $(\mathrm{A}_-\ \boldsymbol{M}\ \boldsymbol{N})$ has a number of rules less than the number of rules of ***N*** if the rule $\boldsymbol{M} \in \boldsymbol{N}$.

□

Proposition 24 *[Non Preserving Effect] An application of any deletion elementary adaptive action may not preserve the original computational function.*

Proof. Directly from the definition 22 and previous lemma 23, since $\mathrm{A}.\boldsymbol{MN} = \boldsymbol{N'}$, and the original sequence of rules $\boldsymbol{N}$ may not be preserved if the rule $\boldsymbol{M}$ was in the sequence. Consequently, if a computation needs the removed rule $\boldsymbol{M}$ to take effect, then it will not happen according to [6, 7].

□

4.3. λ-Adaptive Actions

An adaptive action, as mentioned in definition 7 in the section 2, is defined using the elementary adaptive actions, its semantics is then defined by:

$$\boldsymbol{(\lambda p_1 p_2 \cdots p_n . A_b\ (\lambda g_1 g_2 \cdots g_m .(\lambda v_1 v_2 \cdots v_p .(M_I(\lambda z. M_D(\lambda y. M_A(A_a)))))C)G_1 \cdots G_m)P_1 \cdots P_n}$$

where:

- Each $\boldsymbol{p_i}$, $\boldsymbol{0 \le i \le n}$ is a formal parameter of the adaptive action;
- Each $\boldsymbol{P_i}$, $\boldsymbol{0 \le i \le n}$ is a real parameter value of the adaptive action;
- Each $\boldsymbol{g_i}$, $\boldsymbol{0 \le i \le m}$ is a generator parameter of the adaptive action;
- Each $\boldsymbol{G_i}$, $\boldsymbol{0 \le i \le m}$ is a generator parameter value of the adaptive action;
- Each $\boldsymbol{v_i}$, $\boldsymbol{0 \le i \le p}$ is a variable of the adaptive action;
- $\boldsymbol{C}$ is the *Constraint set* for the inspection elementary adaptive actions;
- $\boldsymbol{M_I}$ is the set of inspection elementary adaptive terms;
- $\boldsymbol{M_D}$ is the set of deletion elementary adaptive terms;
- $\boldsymbol{M_A}$ is the set of insertion elementary adaptive terms;
- $\boldsymbol{A_b}$ is the adaptive action called before the execution of the current action;
- $\boldsymbol{A_a}$ is the adaptive action called after the execution of the current action;

The order of execution of the actions is explicitly defined and follow definition 1, but inside each set of elementary actions there is no order to follow, they execute in parallel.

Consider the definition of new names, as mentioned at the beginning of section 3, using λ-calculus it is possible to define a function that can express the new names as a cartesian product between a finite set of variable names (previously defined), for instance A and the set N. The function *genName* : $\phi \to A \times \mathrm{N}$ actually generates a new name based on a variable name from the set A. It is also possible to define this function as a one argument one, as *genName* : $A \to A \times \mathrm{N}$.

All the issues that appeared in section 3.4 still remains to be treated. Those concerns were not covered by the λ-calculus approach, however the semantics of the adaptive actions became more clear and less ambiguous. So it seams to be a nice way to completely define the semantics for adaptive devices to use a typed version of the λ-calculus. This way the issues raised in section 3.4 can be dealt properly.

5. Structural Operational Semantics using typed terms

The way we used to describe the terms is entirely based on the theoretical definitions of adaptive devices proposed in section 1 and [7]. The adaptive functions do not return values, so their counter-domain is ϕ. The possible existing types are defined in the function's body, and they separate the syntactical elements of the function. In order to keep things following this path, we will assign types only inside the adaptive functions, to allow them to be able to distinguish between variable and parameter types, and, doing so, become aware of type-mismatches and other errors.

To assign types to a non-typed language is a well studied task, and we can use the algorithm **W** proposed by Damas and Milner [3, 8]. In the algorithm the types are defined based on constant types and the type variables, also the function types are defined but we will not need them. And then the type schemes and environments are defined over the types. Doing so, we have:

Type	τ	::=	ι	constant type (*rule, event, function*)
		|	α	type variable
		|	$\tau \rightarrow \tau$	function type
TypeScheme	α	::=	$\tau \mid \forall \vec{\alpha}.\sigma$	
TypeEnv	Γ	$\in$	$V_{ar} \rightarrow^{fin}$ *TypeScheme*	type environment

The constant types introduced were rule, event, function. These types are the primary ones and they can be used to verify syntactical errors in the adaptive functions. Also the parameters must be type-checked, but there are no types in the formal parameters definition. So in the function call the type checker must define the actual parameters type before calling the adaptive action. When the adaptive actions have some arguments passed in their calling process, the types of the arguments were assigned before the control is passed to the function (adaptive action).

Consider an instance of a type **T**, denoted as σ**T**, a constant denoted by C, a function domain as *DOM*(*f*), then:

$$\sigma(\mathrm{C}) = \mathrm{C}$$

$$\sigma(X) = \begin{cases} T, if\ (X \mapsto T) \in \sigma \\ X, if\ X \notin DOM(\sigma) \end{cases}$$

$$\sigma(T_1 \rightarrow T_2) = \sigma T_1 \rightarrow \sigma T_2$$

$$\sigma(x_1{:}T_1, \ldots, x_n{:}T_n) = (x_1{:}\ \sigma T_1, \ldots, x_n{:}\ \sigma T_n)$$

$$\sigma_\circ\gamma = \begin{cases} X \rightarrow \sigma(T), each\ (X \mapsto T) \in \gamma \\ X \rightarrow T, each\ (X \mapsto T) \in \sigma, X \notin DOM(\gamma) \end{cases}$$

$$(\sigma_\circ\gamma)\mathrm{S} = \sigma(\gamma\mathrm{S})$$

With these definitions the issues that appeared in section 3.4 can now be treated, because one can impose types and check them. Any kind of problem, for instance to keep a model deterministic for any operation, can be treated using this approach, by considering the way the *deterministic type* can be kept, so the type-checker may be able to determine the correct type of a model.

To follow this path there is a need for the definitions of the all possible types and their extensions. These definitions have to be done before applying them to the algorithm **W**, and some of the types defined may need specific algorithms for the type-checker.

Consider the following non-exaustive list of types to be formally defined:

- deterministic or not;
- connected (graph) or not;

- preserves previous computational properties or not;

An extension immediately appears as an algorithm and a type, the identity of the model or machine. Any model must be univocally identified.

The research conducted yet is far from complete, but gave the directions for its closure.

6. Conclusion

This paper has started a path proposing a structural operational semantics for adaptive devices. At first, in section 3, the operational semantics were defined directly from the definitions illustrating that a great effort must be spent in order to achieve a fair definition for the semantics. There is another way to define it, in terms of some other formalism, such as, the pure λ-calculus semantics, as shown in section 4. But all the issues of the adaptive devices semantics may not be solved by those two approaches.

The results achieved in this paper have shown that a formal semantic analysis of the adaptive devices is a true necessity. It is not an academic exercise, because of the complexity and difficulty to understand what goes behind the adaptive actions, what sort of problems can emerge, how it can be effective and improve the performance of a device.

Even using the semantics of the λ-calculus it may not be possible to address adequetely some of the problems that arose on the first attempt. Hence, the definitions adopted have shown that an appropriate way to define semantics for adaptive devices may be to use a typing system. This way several non addressed problems can be treated, such as, the distinction between values inside terms, the question about non-deterministic behavior, the insertion of duplicated rules, the insertion of useless rules, the question of unbounded increasing ordinary devices, the fail processing of the ordinary device, the deleted rules may turn some parts of the ordinary device unreachable, the device may behave abnormally.

Introducing a typing system and using algorithm **W** is a nice way of dealing with those problems, but this research has just introduced a path to be followed. There are also other ways to address the semantics of the adaptive devices.

The research conducted must follow this path and introduce a typing system to address these issues. Another path is to use another basis for the semantics, such as category theory, or second order typed λ-calculus, and formulate the adaptive mechanism using λ-terms.

References

[1] Henk Pieter Barendregt. *The Lambda Calculus Its Syntax and Semantics*. North-Holland – Elsevier Science Publishers, Amsterdam, The Netherlands, 2nd edition, 1984.

[2] Henk Pieter Barendregt. Lambda Calculi with Types. in *Handbook of Logic in Computer Science*, volume II, pages 117–309, 1992.

[3] Luis Damas and Robin Milner. Principal type-scheme for functional program. In *POPL 1982: Proceedings of the 9th Annual ACM Symposium on Principles of Programming Languages*, pages 207–212, ACM. New York, 1982.

[4] N. G. de Bruijn. Lambda Calculus notation with nameless dummies, a tool for automatic formula manipulation. *Indagationes Mathematics*, 34(1):381–392, 1972.

[5] Michael J. C. Gordon. *Programming Language Theory and its Implementation*. Prentice-Hall International (UK) Ltd, London, Great Britain, 1st edition, 1988.

[6] João José Neto. Adaptive automata for context-dependent languages. *ACM SIGPLAN Notices*, 29(9):115–124, 1994.

[7] João José Neto. Adaptive rule-driven devices - general formulation and case study. In *CIAA 2001: Proceedings of the 6th International Conference on Implementation and Application of Automata - Lecture Notes in Computer Science*, volume 2494, pages 234–250. Springer-Verlag, 2002.

[8] Oukseh Lee and Kwangkeun Yi. Proofs about a Folklore Let-Polymorphic Type Inference Algorithm. *ACM Transactions on Programming Languages and Systems*, 20(4):707–723, 1998.

[9] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Dept., Aarhus University, 1981.

[10] J. Alan Robinson. Computational logic: The unification computation. *Machine Intelligence*, 6(1):63–72, 1971.

[11] Ricado Luis de Azevedo da Rocha. Adaptive Automata Limits and Complexity in Comparison with Turing Machines. In *Proceedings of the First International Congress of Logic Applied to Technology - LAPTEC'2000 (in Portuguese)*, volume 1, pages 33–48. Faculdade Senac de Ciência e Tecnologia, 2000.

[12] Ricado Luis de Azevedo da Rocha. A Structural Semantics Approach to Adaptive Devices. In *Proceedings of the VI International Congress of Logic Applied to Technology - LAPTEC'2007*, volume 1, pages 1–8, 2007.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-28

Temporal Logic Applied in Information Systems

Silvia RISSINO [a], Germano LAMBERT-TORRES [b] and Helga G. MARTINS [b]
[a] *Federal University at Rondonia Foundation*
BR 364, Km 9,5 – Porto Velho – Caixa postal 295 - 78900-500 – RO - Brazil
[b] *Itajuba Federal University*
Av. BPS 1303 – Itajuba – 37500-903 – MG – Brazil

Abstract: This chapter presents a revision of logic temporal, where the potentialities and used in temporal logic in information systems are presented, mainly what refers to its development. The purpose of this work is to show how Temporal logic is applied to an Information System besides introduction an elementary introduction of Modal Logic, Kripke Semantics, the features of Temporal Logic and it use in some areas of Computer Science.

Key Words: Temporal Logic Temporal, Modal Logic, Kripke Semantic, Information Systems, Temporal Database, Applications.

Introduction

The term "Temporal Logic" is used to describe a system of rules and symbols that represents reasoning with the presence of time a primary element.

The concept of Temporal Logic was first put forward in the sixties by Arthur Prior under the name of Tense Logic and consequently used by logicians and computer scientists [1].

Temporal Logic includes Computational Tree Logic – CTL, which includes as a subset Linear Temporal Logic – LTL; Interval Temporal Logic – ITL; μ-Calculus which includes as a subset Hennessy-Milner Logic (HML) and early Time of Actions Logic. These same systems also include temporal logic to formalize understanding of philosophical subjects related to time, to define the semantics of temporal expressions in natural language, to define a language that codifies temporal knowledge in artificial intelligence, to act as a tool in the control of temporal aspects of program execution as well as being the toll used the construction of queries to temporal database used in systems historical information.

This chapter shows the fundamental concepts of logic paying special attention to the temporal logic, it gives a basic introduction of modal logic and semantics of Kripke, it presents the concepts of system of information and database in addition to the characterization of Temporal Logic and uses in several areas of Science with an emphasis a information systems.

1. Origin, Definition and Classification of the Logic

The word Logical comes from the Greek word Logos, which means reason or action of reason. Logical it is the knowledge of reason a actions of reason [2]. As a science logic defines the declaration structure and argues that it elaborates formulas through which these can be codified.

Logic is a science of tendencies and characteristic of Mathematics, strongly coupled to Philosophy that ensures the rules of logic think [3]. The learning of logic does not constitute an end in itself and only applies when used in the guarantee of the thought being structured correctly in order to arrive at true knowledge.

Logic can be characterized as the study of principles and inference methods or reasoning. Logic always uses the same basic principles of Classic Logic: the law of identity, no contradiction and of excluded middle. Besides Classic Logic other types of logic can be utilized depending upon the context.

Logic can be divided into two categories; Inductive used in the theory of probability, and Deductive, which can be classified as either classic logic, complementally Logics of Classic Logic and the Non Classic Logic.

1.1 Classic Logic

The Classic Logic is considered the cornerstone of deductive logic and forms part of the calculus of first-order predicates together with identity and functional symbols. It can be characterized by a consequence relation defined syntactically and semantically [4]. The classic logic is based on three principles:

- **Principle of Identity**
  The idea that every object is identical;

- **Principle of Contradiction**
  Given that two propositions are contradictory one is of them is false;

- **Principle of excluded middle**
  Given that two are contradictory one of them is true.

1.2 Complementally Logic of Classic Logic

The Complementally Logics of Classic has besides the three principle mentioned above, three principles that govern it thus, extending it's domain:

- **Modal Logic**
  Adds to the classic logic the principle of the possibilities;

- **Epistemic Logic**
  Also know as knowledge logic adds to classic logic the principle of uncertainly;

- **Deotonic Logic**
  Adds to classic logic the principle of morals in the form of rights, prohibitions and obligations.

1.3 Non Classical Logic

Non Classical Logic is characterized by the non use of same or all principles of classic logic [2]:

- **Paraconsistent Logic**
  It is a logic form of logic where by the principle of contradiction does not exist. In this logic type as many affirmative sentences as true or false depending on the context [5].

- **Paracomplete Logic**
  This logic does not consider the principle of excluded middle, that is a sentence which be thought of as completely true or false.

- **Logic Fuzzy**
  Also known as diffuse logic, it works with the concept of pertinence degrees. As well as the logic paracomplete, it does consider the principle of excluded middle, but rather one of comparative way using the element fuzzy set. This logic form has wide applications both in computer science and statistics as well as forming base for indicators make up Index of Human Development [6].

- **Temporal logic**
  It is widely used to represent the temporal information of a logical structure.

2. Elements of Modal Logic

First-Order Logic is limited in the sense that it does not distinguishing between the concept of Possibility and Necessity. According to the classical formalism, formulas are just true or false, without any kind of qualification. Modal Logic permits the representation the concepts of necessity and possibility, that is, it represents the study of the behavior of the expressions of necessity (it's necessary that) and possibility (it's possible that). These concepts are formalized from the notion of Possible World whose interpretation may be described as a conceivable alternative of the real world.

The modal logic emerges within the context of Temporal Logic, as it language contains, besides functional truth operators, of modal logic that are: Necessary □ and Possibility ◊.

The syntax and semantics of Modal Logic are derived from first-order logic. Temporal Logic specifies its functions from the semantics of modal logic by adding the time factor [8]. The quantificators of modal logic are the Necessary □ and Possibility ◊, where:

Being that ◊ϕ is possible that ϕ is true;

Begin that □ϕ: is necessary that ϕ. is true;

Given that □ϕ→◊ϕ: everything that is necessary is possible;

Given that ϕ→◊ϕ: if anything is true, then it is possible;

Given that ϕ→□◊ϕ: something that is true is necessarily possible;

Given That ’ϕ→"’ϕ: everything that is possible is necessarily possible.

3. Kripke Semantic

Kripke semantic is a class Kr of Kripke models, where the system K is considered to be the smallest of the normal modal systems. That is, it is the intersection of all the normal modal systems, justified by following principles: it is modal logic systems, with a set of axioms and inference rules that represent that reasoning formally. Given that P is a set of atomic propositions, therefore a Kripke frame is a tupla (S, i, R, L), where [9]:

- S is finite set of states;
- $i \in S$ is the initial state;
- $R \subseteq S \times S$ it is a relation of total transition: $\forall s \in S \,.\, \exists s' \in S \cdot (s, s') \in R$
- L: S $\rightarrow \wp(P)$ it is a function that identifies each state with the set of valid atomic formulas inside that state.

The semantics of Temporal Logic may be defined on Kripke structure, thus, the specification techniques and the verification of the properties may be presented regardless of the formalism of the model to be employed.

4. Temporal Logic

4.1 An Outline of Temporal Logic

Classical logic is a kind of “true truth”, in the sense that it does not have any mechanism that represents time and its consequences on the truth values in the logic formulas.

Temporal Logic has as its base on Modal Logic and permits the representation of varying states within varying periods of time. It allows us to verify the authenticity of the assertions over time. Since an assertion can be true in particular instant of time and false in another.

For the problems of interest in Artificial Intelligence and Information Systems, where time is of the utmost importance, since they need a representation of events and their sequences in time in order that they can be resolved efficiently [7]. As an example, one can cite the problems in the area of medical diagnostics, understanding of “medical stories” or even problems in engineering as a whole.

In the approach of Temporal Logic the properties to be verified are given through the Kripke formulas (Possible World Approach) that represent a set of states, of transitions among states and functions that label each state inside with the set of true properties of it.

The main characteristic of Temporal Logic is that of a certain logical formula representing different values of truth within different instants of time. This characteristic is formalized through the introduction of several temporal operators in the syntax of first-order logic language. In this case, there is a Method of Temporal Arguments, where A is any formula, P is an instant in the past, F is an instant in the future, G all the instants in the future and H all the instants in the past, according to Table 1.

Table 1. Methods of Temporal Arguments in Relation to Formula A.

PA	$\exists t\,(t<now\ \&\ A\,(t))$	A was true in some instant of the past
FA	$\exists t\,(now<t\ \&\ A\,(t))$	A will be true in some instant of future
GA	$\forall t\,(A$ do $t<now \rightarrow (t))$	A will be true in all of instants of the future
HA	$\forall t\,(A$ do $now<t \rightarrow (t))$	A was true in all instants of the past

When considering a change of true values over a given time both the interpretations and models of Temporal Logic, should included a structure that represents the instants of time and its precedence relation.

The temporal structure is defined as an empty set $\mathcal{T}$ of instants of time and a precedence relation $\prec$ on the elements of $\mathcal{T}$. Using a temporal structure it is possible to extend the association function $\mathcal{E}$ together with the semantics of logical operators of first-order logical, so that transcend dependency on an instant of time $t \in \mathcal{T}$. In order to complete the semantics of the Temporal Logic one should to add the relative rules to the temporal operators:

- FA has true value T in instant $t \in \mathcal{T}$ if exist $t' \in \mathcal{T}$ such that $t \prec t'$ and A has value T in t’;

- PA has true value T in instant $t \in \mathcal{T}$ if exist $t' \in \mathcal{T}$ such that $t' \prec t$ and A has value T in t’;

The operators G and H can be defined through the equivalence relationship:

$$\mathbf{GA} \leftrightarrow \neg\mathbf{F}\neg\mathbf{A}\ \mathbf{e}\ \mathbf{HA} \leftrightarrow \neg\mathbf{P}\neg\mathbf{A}$$

A formula is true in Temporal Logic if it represents truth value T at all time instants. As in the modal logic, the properties of temporal precedence relation determine the characteristics of several temporal logics. The precedence relation without any restriction corresponds to minimum temporal Logic K, which may be characterized by the inference rule *Modus Pones* and the following set of axioms showed in figure 1 and 2, where the linearity are presented backward and forward.

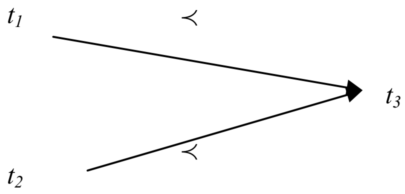


Figure 1 – Linearity forward.

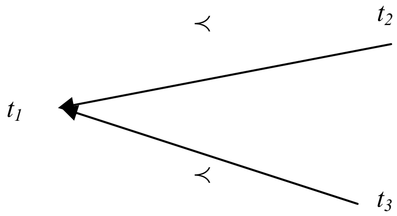


Figure 2 – Linearity backward.

If A is Tautology, then:

$$G(A \rightarrow B) \rightarrow (GA \rightarrow GB)$$

$$H(A \rightarrow B) \rightarrow (HA \rightarrow HB)$$

$$A \rightarrow HFA$$

$$A \rightarrow GPA$$

If A is Axiom, then $GA \Lambda HA$.

The first restriction concerning the precedence relation demands that the time be decision tree, that is, which has existed a "past", although the future remains open. This restriction can be formalized thus:

$$\forall t1, t2, t3 \in \mathcal{T}, ((t1 \prec t2) \Lambda (t2 \prec t3)) \rightarrow t1 \prec t3$$

$$\forall t1, t2, t3 \in \mathcal{T}, ((t1 \prec t2) \Lambda (t2 \prec t3)) \rightarrow ((t1 \prec t2) \vee (t2 \prec t1) \vee (t1 = t2))$$

The first condition is transitive, due to precedence in time. The second condition is known as lineal backward, that impedes the instant of related time, as show in figure 1. The temporal Logic associated this restriction is characterized by the axioms logic K, as well as the following axioms:

$$FFA \rightarrow FA$$
$$(PA \Lambda PB) \rightarrow P(A \Lambda B) \vee P(A \Lambda PB) \vee P(PA \Lambda B)$$

4.2 Types of Temporal Logic

a) Computational Tree Logic (CTL)

It is branching-time logic, meaning that its model of time is a tree-like structure in which the future is not determined; there are different paths in the future, any one of which might be 'actual' path that is realized. CTL, which includes as a subset Linear Temporal Logic, which is a modal logic with modalities referring to time. In LTL one can encode formulae about the future of paths such as that a condition will eventually be true, that a condition will be true until another fact becomes true, etc.

b) Interval Temporal Logic

It is a temporal logic for representing both propositional and first-order logical reasoning about periods of time that is capable of handling both sequential and parallel composition. Instead of dealing with infinite sequences of state, interval temporal logics deal with finite sequences. Interval temporal logics find application in computer science, artificial intelligence and linguistics. First-order interval temporal logic was initially developed in 1980s for the specification and verification of hardware protocols.

c) μ-Calculus

It is a class of temporal logics with a least fix point operator μ. It is used to describe properties of labeled transition systems and for verifying these properties.

d) Hennessy-Milner Logic

It is a temporal logic in computer science. It is used to specify properties of a labeled transition system, a structure similar to an automaton; it was introduced in 1980 by Matthew Hennessy and Robin Milner.

e) Temporal Logic of Actions

It is developed by Leslie Lamport, which combines temporal logic with a logic of actions, it is used to describe behaviors of concurrent systems.

4.3 Nature of Time in Temporal Logic

a) Linearity of Time

The generally accepted concept of time is that it is a lineal sequence of instants of decision tree. Time, in Newton's classic sense, is in the lineal way. Therefore, the restriction concerning the precedence relation can be formalized by the introduction of a condition of linearity forward:

$$\forall t1, t2, t3 \in \mathcal{T}, ((t1 \prec t2) \Lambda (t1 \prec t3)) \rightarrow ((t2 \prec t3) \vee (t3 \prec t2) \vee (t2 = t3))$$

This condition impedes the existence of instants of related time exist, as shown in figure 2. The two linearity types may be combined into a simpler a condition known as connectivity:

$$\forall t1, t2 \in \mathcal{T}, ((t1 \prec t2) \vee (t2 \prec t1) \vee (t1 = t2))$$

This new Temporal Logic, known as K1 can be characterized by Logic Kb axioms as well as following axiom:

$$(FA \Lambda FB) \rightarrow F(A \Lambda B) \vee F (A \Lambda FB) \vee F(FA \Lambda B)$$

b) Limits of the Time – Start and End

Another important consideration regarding the nature of time is the existence of its limits, that is, the existence or otherwise of an initial time and an end time. The classic physics consider the time limited in terms of past and future, this consideration being formalized by the following precedence relation:

$$\forall t2 \in \mathcal{T}, \exists t1 \in \mathcal{T}, t1 \prec t2$$

$$\forall t1 \in \mathcal{T}, \exists t2 \in \mathcal{T}, t2 \prec t1$$

This new Temporal Logic, known as Ks, can be characterized by Logic K1 axioms as well as the following axioms:

$$GA \rightarrow FA$$

$$HA \rightarrow PA$$

c) Time Density

A lineal series can be presented by the structure of whole, rational or real numbers. The Logic of Ks considers a whole number structure as one, where two instants can exist without a third intermediary.

A dense time is considered as a structure of rational numbers where an intermediate instant exists between two others, it can be formalized by the following condition:

$$\forall t1, t2 \in \mathcal{T}, \exists t3 \in \mathcal{T} \ ((t1 \prec t2) \rightarrow (t1 \prec t3) \ \Lambda \ (t3 \prec t2))$$

The Logic that adopts, is called as Kp, it can be characterized by the axioms of the Ks Logic as well as the following axiom:

$$FA \rightarrow FFA$$

A continuous time with a structure of real numbers can be characterized in the following way: if the set $\mathcal{T}$ is divided into two other set $\mathcal{T}_1$ and $\mathcal{T}_2$, such that $\mathcal{T}=\mathcal{T}_1 \cup \mathcal{T}_2$ and any element of $\mathcal{T}_1$ precedes all elements of $\mathcal{T}_2$ then there must always be an element of $\mathcal{T}$ that result from all the elements of $\mathcal{T}_1$ and precedes all of elements of $\mathcal{T}_2$. This condition can be formalized for:

$$\forall \mathcal{T}, \mathcal{T} \subseteq \mathcal{T}, ((\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2) \ \Lambda \ (\forall t1 \in \mathcal{T}_1, \forall t2 \in \mathcal{T}_2, t1 \prec t2)) \rightarrow$$
$$(\exists t \in \mathcal{T}, \forall t1 \in \mathcal{T}_1, \forall t2 \in \mathcal{T}_2, (t1 \prec t)) \ \Lambda \ (t \prec t2))$$

This condition is absorbed by the following axiom:

$$((GA \rightarrow PGA) \ \Lambda \ G(GA \rightarrow PGA) \ \Lambda \ H(GA \rightarrow PGA)) \rightarrow (GA \rightarrow HA)$$

That it formed together with the axioms the Kp,Logic and the Continuous Temporal Logic, known as Kc.

4.4Forms of Representing the Time in Temporal Logic

Temporal Logic includes Computational Tree Logic which in turn has as a subset Linear temporal Logic; μ-Calculus which includes as a subset Hennessy-Milner Logic (HML) as well as Interval Temporal Logic and more recently the Time of Actions Logic, These logics disagree as tor the way that they represent time. In terms of time these are two basic representation models [9]:

a) Lineal Time: All the behavior of system consists of set of infinite point that it start in the initial state *i*;

b) Branching Time: All the behavior of systems is represented by computation tree of infinite depth in which the root is the initial state *i*.

Both models may be computed using the kripke of structure; however branching time has more information so that properties can be verified using branching time.

In Linear Temporal Logic, the formulas are interpreted on infinite lines, besides the usual connectives of the logic propositional it has the following used temporal operators, in the case of branching time:

Next X f when f is valid in the next state;

Future F f when f is eventually valid;

Globally G f when f is always valid;

Until f U g when f is valid until that g is it;

Release f R g when the occurrence of a state where f is valid frees the being's g.

5. Information Systems

Information Systems is a term used to describe an automated system, it encompasses both people and machines together with/or organized techniques to collect, to process, to transmit and disseminate data that represent information [10]. The term is also used to describe the study area of Information Systems, Information Technology and their application within an organization.

The area of Information Systems is considered by the researchers as a area multi or to trans-discipline, due to the interrelations with other areas, such as Computer Science, Administration, Economy, Sociology, Law, Engineering, Information of Science.

The most up to date definition of Information Systems also consider the Telecommunication Systems and or related equipments; systems an interconnected subsystem that use equipment in acquisition, storage, manipulation, management, movement, demonstration, exchange, transmission, or reception of the voice and/or data besides including the software and hardware.

5.1 Classification of Information Systems

Information Systems, depending in their specificities and use can be classified as [11]: Systems of Operational Information; Systems of Managerial Information; Strategic Information Systems; Geographical Information System Geographical; Decision Support System and Specialist System.

5.2 General Architecture of the Systems of Information

The systems of information in general are basically composed of five components; these are connected in a hierarchy of three levels. The level closest the user is known as the interface man-machine. In the intermediate level, the system of information should include data processing mechanisms for the entrance, edition, analysis, visualization and exit of data. In the internal level of the system, there should be a management system of databases.

The principal main components of the architecture of systems of information can be showed in de figure 3 and described as the following:

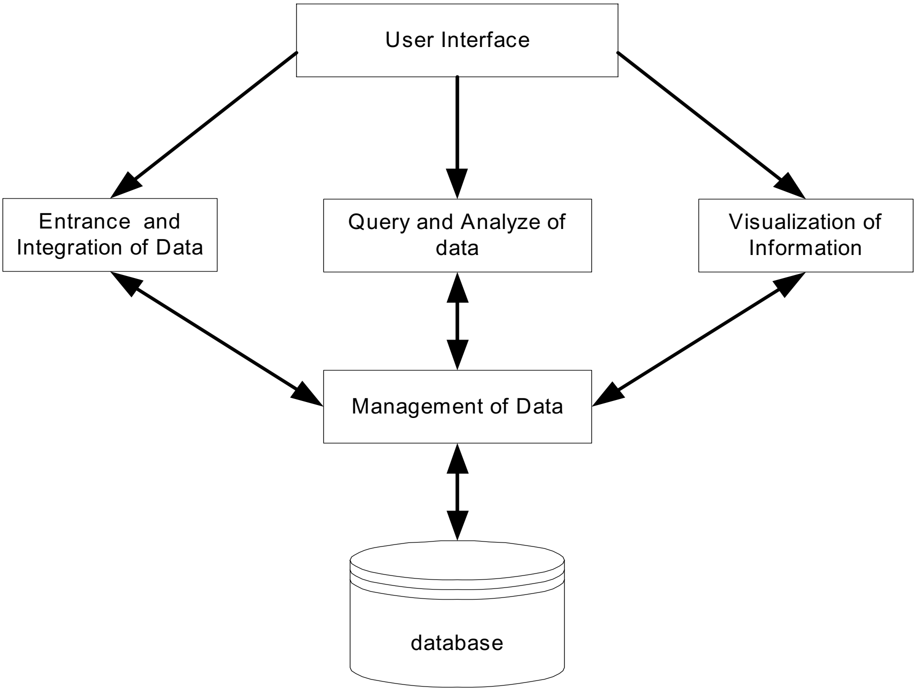

User Interface
Entrance and Integration of Data
Query and Analyze of data
Visualization of Information
Management of Data
database

Figure 3 – general architecture of information system

- User Interface (interface man-machine): where it is defined as the controlled and operated systems;

- Entrance and integration of data: Where defined integrity rules that, guarantee the consistence of the database;

- Query functions and analysis of data: where query is created to transform it into information in this level the developed query, in certain situations, use the temporal logic as support as for its implementation;

- Visualization of Information: where is defined the form of visualization of the information;

- Database - Storage and recovery of data: the kernel of the system of information, as within this component the database, together with the language of manipulation of data and with the whole structure, access rules, safety and disponibility of data.

The components of S.I. they link in a hierarchical way thus enabling the managers of organizations develop their work practices with confidence providing a more efficient administration.

6. Database

A database is a structured collection of records or data. A computer database relies upon software to organize the storage of data. The software models the database structure in what are known as database models [12].

The computation community, to indicate organized collections of data stored in digital computers, created the term database initially; however the term is used now to indicate as much digital databases as available databases in another way.

Database management systems (DBMS) are the software used to organize and maintain the database. These are categorized according to the database model that they support. The model tends to determine the query languages that are available to access the database. A great deal of the internal engineering of a DBMS, however, is independent of the data model, and is concerned with managing factors such as performance, concurrency, integrity, and recovery from hardware failures.

6.1 Database Classification

The most practice method of classifying databases is that that utilizes the considerations of the user this is a data model.

Over the years several models been used each with its advantages and disadvantages. However today, the most common classification can be described as the following:

- **Navigational Model**

It proposed in sixties as computers with more storage capacity were becoming an integral part of company expenditure two navigational models were developed: Network Model (CODASYL - Committee for Data Systems Language) and Hierarchical (IMS. Information Management System) Model.

- **Relational Model**

It developed by Edgar Frank Codd in early seventies, initially as a mathematical theory and in a shat space of time wan over the confidence of manufactures of database [13]. In this model, the logical structure of database is independent of the method of physical storage. This system has since became standard principally because maintenance is facilitated as the logical structure is separated from the storage method and witch in turn facilitates the insert or retraction of attributes of the logical model without the need of rewrite the application code.

- **Entity-Relationship Model (ME-R)**

It proposed in 1976 by Peter Chen, for database projects and gave a new and important perception related to the concepts of data models [14]. ER modeling makes it possible for the planner to concentrate solely on data use without worrying about the logical structure of tables. This model consists of mapping the real world of the system into a graphic model that will demonstrate the model together with the existing relationship between data.

- **Model Oriented-Objects**

The techniques orientation objects which was developed in the late sixties has, since the early nineties been used is databases and has created a new programming model known as database oriented objects and has became increasingly popular in the design and implementation of systems [15].

- **Relational Extended Model or Object-Relational**

It was developed during the nineties because of the need for applications use certain oriented-objects characteristics, using earlier break wags of model models relational research [16].

- **Semi-Structured Model**

It represents a new paradigm in database that is different to relational and oriented-object paradigms. The outline of semi-structured data is built into the data making it extremely useful in the exchange of data between applications and organizations [17].

- **Model Space/Geographical**

It is used as tools to treat data in treatment of data in geoprocessing applications, where data originates from sources digital [18].

- **Model Temporal**

It consists of model for database applications that represent some aspect of time when organizing information. In a temporal database time is considered as an orderly sequence of points with same granularly that depends upon the application [19]. The importance of the temporal model database is to be found in the modeling and development of a database that is to say the stored data obey the temporal characteristics that are evident in the following applications:

- Registrations of academic information;
- Electric systems;
- Financial applications;
- Hotel reservations;
- Registration of hospital patient records.

Recent research carried out in the area of temporal database has had as one of its objectives the definition of concepts and strategies used in the treatment of historical data.

When data process temporal characteristics, there is a need to register the data\inclusion procedure that in turn generates a demand for exploring aspects related to the sequence of stored data.

The modeling of temporal data is necessary in order far the representation of the dynamic of applications as well as the representation of temporal interaction among the various processes.

6.2 Management of Database Systems

Management of Database Systems is a group of programs that allow the user to manipulate the database. Management of Database Systems were created so as to manipulate large amounts of data and to offer data resources such as [20]: Control of Redundancy; Integrity; Consistence

6.3 Time Dimension Databases

A database is a set of secure registrations that is saved systematically storage into a computer so that it may be query by a particular program [19]. A registration is usually associated with a complete concept and is divided into fields or attributes that give valves to the properties of these concepts. Same registrations may appear directly or reference indirectly to other registrations which are part of the general character of the model adopted by the database [20].

The relational model has become popular largely because of its simplicity of use and its solid mathematical base. The version of relational model initially proposed by Codd in 1970, did not in the temporal dimension of data instead the variation of data was treated over a space of time in the same way as ordinary data and was not appropriated in the application that required past, present and/or future data values. Today most applications require temporal data [21].

a) Data Modeling in Information System

As the dynamic characteristics of applications are presented through temporal aspects. The model used in development of the information systems should allow far all of the temporal aspects to be id identified a presented. A database that possesses the capacity to store data from the past, present and future is known as a temporal database and differs from the conventional model in the type of model used as well as its query language.

The modeling of time is a very important consideration when modeling information systems.

Management of Database Systems is systems that both represented and manage application data. Far these systems to represented scenarios that are as close as possible to the real world it is necessary to introduce a time dimension. In a conventional database reality is presented only is its present state. When the real world changes, the new values are incorporated into the database so substituting the old ones [22].

In temporal database, when the real world changes, a new factor is inserted into the database. It is, therefore necessary for this new factor to be considered besides any already existing.

b) Representation Information in Temporal Database

The implementation of the concept of time, in temporal database, can be accomplished in three ways:

- The manipulation of temporal data is accomplished explicitly by the user;

- The manipulation of temporal data is accomplished through actions associated with defined proprieties such as temporal and corresponding to be semantic extensions related to normal data;

- The temporal properties are treated as an extension of both the data model and manipulation language.

c) Time: Concepts and Definitions

The time possesses a relative concept; it depends on the situation in that is presented. With the introduction of factor time, to be treated by applications in Management of Database Systems, was necessary to increase to database concepts the aspect of temporal of data. In this section, the concepts related and derived of the time are presented, and the presented definitions are based on the work of Jansen et al 1998 [23].

↔Time

The relative concept at time is indicated by intervals or duration periods, other form of defining time is through the event definition, that can be said that happened after other event, because, the can be measured as an event happens after other. The all is the amount of time among two events, being the separation of the two events an interval and, the amount of that interval is the duration.
The difficulty of defining the time takes the researchers to seek the best form of temporal representation, that is, to define all of the elements that are involved in the definition to avoid conceptual unconsciousness. When it is if manipulating information in database storms is taken into account the time of transaction, time of validity and the time defined by the user:

- **Transaction Time**

  A database fact is stored in a database at some point in time, and after it is stored, it is current until logically deleted.
  The transaction time of a database fact is the time when the fact is current in the database and may be retrieved. Consequently, transaction times are generally not time instants, but have duration.

Transaction times are consistent with the serialization order of the transactions. They cannot extend into the future. Also, as it is impossible to change the past, (past) transaction times cannot be changed. Transaction times may be implemented using transaction commit times, and are system-generated and-supplied.
While valid times may only be associated with \facts," statements that can be true or false, transaction times may be associated with any database object;

- **Valid Time**

The valid time of a fact is the time when the fact is true in the modeled reality. A fact may have associated any number of instants and time intervals, with single instants and intervals being important special cases. Valid times are usually supplied by the user;

- **User-Defined time**

User-defined time is an uninterrupted attribute domain of date and time. User-defined time is parallel to domains such as \money" and integer- unlike transaction time and valid time, it has no special query language support. It may be used for attributes such as \birth day" and \hiring date."

↔ **Order Time**

In models of data in that there is need to represent a temporal data, it is done necessary to the definition of an order in the time, so that it can represented in a lineal way the information that will originate from the data that was stored.
This to mean to say that is necessary to define which the interval that it will work (example: beginning Time T and Time end T', i.e. T<T', therefore the events that originate the data in T should be stored first than the data originated in T′.

- **Variation time**

The variation of time has for objective to represent the sequence of time to be used; it can be in two ways:

- **Continuous time**

The time is continuous by nature, because it representation comes from the own nature;

- **Discreet time**

Discreet time is based on the existence of a line of time that is composed by sequential consecutive of temporal intervals, where there is no possibility of decomposition of these intervals. This interval or unit of indivisible time is known of chronon and it has identical duration. There are other forms of discreet variation:

- **Variation for events**

Where the value of the data is constant from it definition to the moment / instant that other value is defined for the data in function of the occurrence of an event. This variation type is very common in the applications that use model of temporal data;

- **Variation point to point**

The defined value is valid only in the temporal point where the same was defined

↔Chronon

In a data model, a one-dimensional chronon is a non-decomposable time interval of some fixed, minimal duration.

An n-dimensional chronon is a non-decomposable region in n-dimensional time. Important special types of chronons include valid-time, transaction-time, and bitemporal chronons.

↔Granularity

The temporal granularity is a parameter that corresponds to the duration of a chronon, that is, to smallest unit of time supported to define a data in a system database manager.

The data can assume a chronon in the day/hour/minute or something similar, but that has as objective the best representation of reality. In spite of chronon of a system to be only, it is possible to manipulate different granularity through functions and available operations in MSDBs that implement the temporal database.

↔Instants

An instant is a time point on an underlying time axis. When the time is continuous, an instant is a point in the time of infinite duration.

As in the real numbers it exists at least a number between two sequential numbers, between two points in the time there will be always another point in the time. In the case the instant of time to be discreet, the instant is represented by one of chronons of the line of time supported by the model.

The discreet time is considered as special instant by representing the current instant, and the time will move in the line of the time to facilitate definition of what is passed, present and of the future.

↔Interval

A time interval is the time between two instants. In a system that models a time domain using granules (defined in the addendum), an interval may be represented by a set of contiguous granules:

- **Open**

  Limits do not belong to the interval. Example:]2,4[, instant initial>2 and instant and<4;

- **Open/Close**

  One of the intervals is inside of the own interval, that is, the limit is made by the own value. Example: [2,4[, instant initial>=2 and instant and<4; or]2,4] instant initial>2 and instant end<=4;

- **Close**

  the limits of the interval are part of the own interval, that is, the intervals are limited for them same. Example: [2,4] instant initial>=2 and instant and<=4.

The temporal interval can be characterized as current instant when the variation of the time is null, i.e., the initial and final times are same.

↔ Limits Time

The limit of the time varies in function of type of the adopted temporal representation, that is, the point in time or the point of reference, and the point in time has a predecessor and a successor. In the models of based data or oriented to objects, referred it concept is defined as time of life of the object, in the case of temporal models that it use this concept, it means that a certain point of the time is part or not of the limit of the interval.

↔Event

Event is a occurrence instantaneity that represents some fact, action or satisfied condition.

↔Span (Duration)

A span is a directed duration of time. Duration is an amount of time with known length, but no specific starting or ending instants. For example, the duration \one week" is known to have a length of seven days, but can refer to any block of seven consecutive days. A span is either positive, denoting forward motion of time, or negative, denoting backwards motion in time.

The representation and the definition of the duration of time depend on the context in that is inserted, therefore it can be fastens or variable:

- **Fastens Duration**

  Certain time of duration in the model, through a constant unit, that serves as rule for the existence and consistence of the data Example: duration of the time used in hours, that is, one hour will always have sixty minutes;

- **Variable duration**

  Time of duration will depend on the context, being possible it change in elapsing of time (variation time). Example: Use of unit of duration of time in months, where the amount of days in every month can varies in the interval [1,31], having situations in that the month has thirty days, twenty-nine days or thirty and one days.

↔Representation Time

The definition of time can be realized in way explicit or implicit in a database. Explicit forms of you exemplify her is through the association of an instant of time to information (timestamp), or in implicit way using temporal logic as well as the following:

- **Timestamp**

  Is a time value associated with some object, e.g., an attribute value or a tuple. The concept may be specialized to valid timestamp, transaction timestamp, interval timestamp, period timestamp, instant timestamp, bitemporal-element timestamp, etc;

- **Lifespan**

  The lifespan of a database object is the time over which it is defined. The valid-time lifespan of a database object refers to the time when the corresponding object exists in the modeled reality. Analogously, the transaction-time lifespan refers to the time when the database object is current in the database. If the object (attribute, tuple, relation) has an associated timestamp then the lifespan of that object is the value of the timestamp. If components of an object are time stamped, then the lifespan of the object is determined by the particular

↔ Relationships Time

In database, the existent relations between the objects are defined in function of some parameters, and one of them is the time, for that the relationships can be:

- **Snapshot relationship**

  Relations of a conventional relational database system incorporating neither valid-time nor transaction-time timestamps are snapshot relations;

- **Valid-Time Relationship**

  Valid-time relation is a relation with exactly one system supported valid time. There are no restrictions on how valid times may be incorporated into the tuples; e.g., the valid-times may be incorporated by including one or more additional valid-time attributes in the relation schema, or by including the valid times as a component of values of application-specific attributes.

- **Transaction Time Relationship**

  A transaction-time relation is a relation with exactly one system supported transaction time. As for valid-time relations, there are no restrictions as to how transaction times may be incorporated into the tuples.

- **Bittemporal Relationship**

  A bitemporal relation is a relation with exactly one system supported valid time and exactly one system-supported transaction time. This relation inherits its properties from valid-time relations and transaction-time relations. There are no restrictions as to how either of these temporal dimensions may be incorporated into the tuples.

d) Time Factor in Databases

A temporal database stores several states of data, as well as the instants in that these different states are valid. In agreement with the type of used temporal label, the following categories of databases can be identified [24]:

↔Snapshot Database

They are the databases, where the only existent values are the current ones. Each modification in the value of a property can be noticed as a transition of database. In a transition, the new fence in substitutes the value previously stored. The current state of database, composed by the current values of all of properties is the only existent;

↔Transaction Time Database

This database type, where the defined value is associated to the temporal instant in that the transaction was accomplished, under form of a temporal label.

A relationship that uses this approach can be seen as had three dimensions: tuplas, attributes, time of transaction.

The databases of time of transaction allow the recovery of defined information in some instant of past, because all the last data associates are stored to his/her definition (time of transaction) instant;

↔ Valid Time Database

It corresponds at the time that information is true in the real world. It is associated the information the valid time, and the user should supply this time. This database type registers the relative history to the data and not to the transactions, allowing the correction of mistakes through the modification of data.

The databases of valid time admit, in its context, intervals of time no valid, and they allow the recovery of valid information in passed moments, presents and futures;

↔ Bittemporal Database

In this type, the database combines the properties of databases of time of transaction and of valid time, i.e., it treats the two dimensions of time. The whole history of database is stored, being possible to have access the all of the last states of database, so much the history of transactions accomplished as the history of validity of data.

The current state of database is constituted by the values now valid. Future values can be defined through the valid time, being possible to recover the moment in that these values were defined for eventual alterations;

e) Temporal Domain in Relational Database

In the last years, several models of temporal data were proposed, where it is evident the scientific community's concern with the temporal representation in the modeling of data [25]. In this work it is used the relational model and their extensions by being the model more used in the implementations of systems of information.

In the systems of relational databases, the data are stored in tables of two dimensions, where the lines are the tuplas and the columns the attributes [12]. When introducing the factor time, associate to the data, the tables can be analyzed as own one more dimension - the time. The figure 4 presents the diagram with three dimensions of a table (tupla, attribute and the time).

Implementation of temporal databases, two approaches exist. The first is to extend the semantics of relational model for incorporation of factor time; the second approach implements the temporal database on the relational basic model with the time appearing as additional attributes.

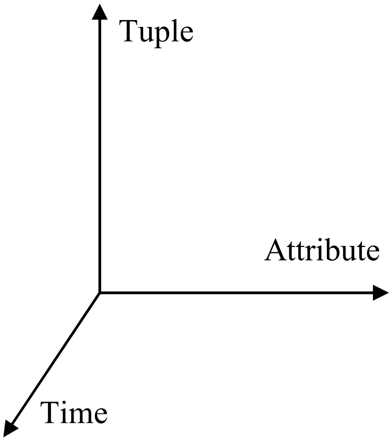


Figure 4 – Diagram with three dimensions of a table.

The implementation of data models that adds the dimension time has been accomplished, most of time, using the option of extending to the models already relates existent the component time. Following it is shown the extended models relate, that use the component time with one of factors of organization of information:

↔HRDM - Historical Relational Data Model

This model is an extension of relational model, where it is incorporate the factor time the usual (attribute and value) dimensions.
In this model it is possible the definition of data that it vary in agreement with the time, grouping storms, where it can be determined the limit and the temporal evolution of outline, since HRDM uses the valid time for the temporal representation.

↔HSQL – ***Historical Structure Query Language***

The model HSQL is oriented to states of historical database, that is, instead of working with the dimension additional time, this model works with the possible states that the database can present. In this model, exist an object called event that just persists for a certain unit of time exists, in case the granularity (unit of time) is certain in days, the events that happen in hours, they are treated as happening at the same time.

The objects of type and the one of type events are represented through historical relationships. A historical relationship is defined by group of attributes. The language of definition of data is in the way of an extended SQL.
The time of transaction is represented, therefore it is destined to the modeling of systems of real time, us which the time of transaction usually corresponds to the of validity [26].

↔TRM - ***Temporal Relational Model***

TRM incorporates the temporal semantics of real world to a model of given relational. Together with its query language TSQL (Temporal Structure Query language), the model allows the manipulation as much of temporal information as of information no temporal, in a coherent and solid way.

Combining the available resources in the model TRM with additional information related to the version of outline, it is possible not only to model an application considering the temporal aspects, as well as to implement the versionament of outlines in this application.

A temporal database is defined as the union of two groups of relationships: the statics and the temporal. In this model, all temporal relationship possesses two obligatory attributes of time: time at beginning and (Ts) time of end (Te). These attributes correspond to the inferior and superior limit of an interval of time, where the temporal relationships represent the valid time [19].

6.4 Temporal Integrity Constraints

The term integrity constraints, says respect not only to the concept for which the rules are created for updating of data of a database, but also to the several aspects related to the form of application of concepts and that the integrity of database is assured. For so much, some aspects are considered, as the characteristics related to the definition and the specification of restriction, besides details of model of data adopted.

The consistence of a database is guaranteed through the definition of rules, which control the integrity and the evolution of data. According to the type of used database, the rules are treated in differentiated ways; therefore each one possesses different characteristics and different focuses. Databases that provide means for creation and execution of rules are usually knew as database actives [27].

When if used temporal database, the rules owe, additionally, to treat of implicit temporal information; the manipulation of data (insert, removal and updating) in a temporal database is, without a doubt, the concept that joins larger value in complexity level [28]. This fact happens due to the different forms with which the information should be treated, according to the time in that they meet. The success of management of data is assured by specific rules of integrity and transitions of states added to the manipulation of times of validity and transaction.

In this section it shows how temporal logic can be used the language goes specifying temporal integrity constraints in relational databases. Here, it discuss different notions of temporal constraint satisfaction and various approaches to the problem of temporal integrity maintenance.

↔ **Constraints Satisfaction**

Temporal integrity constraints are imposed on the history of database, i.e., the sequence of states up to current one. In practice, the satisfaction restriction in a temporal database can be expressed through the maintenance of main characteristic of database - integrity of data. It does not advance to update a database, historical or no, without the warranty of integrity, for that, the satisfaction restriction is directly related this characteristic, what carts the existence of a true (true) state of database. State integrates, in the moment of updating.

↔Temporal Integrity Constraints

According to Chomicki and Toman [29], are three different scenarios of temporal integrity maintenance:

- **Constraint Checking**

  The updating in the database should produce a new solid state for the database, in the case the checking to find some inconsistency type, during the updating (update) process, the new state of database is committed, being necessary the same to be aborted, otherwise the new it is recorded;

- **Temporal Triggers**

  The triggers are known as trigger by the database implementation, it function is to activate functions developed by programmers, that satisfy some condition type demanded for the existence of database. Usually those "programs" are used in the moment of execution of tasks, mainly, those related to the integrity warranty of any database, besides and mainly in the temporal;

- **Transaction validation**

  A transaction is a unit that aims at to preserve the consistence of database, through a group of procedures that it is executed in a database, in which the user just notices as a single action. The integrity of transaction and consequently, of a database, it depends on four such private properties as Atomicity, consistence, isolation and durability. In database temporal it behavior is identical the any other database type, because the warranty of transaction valid not only it is related to the valid time, but there is need to be concluded with success, otherwise the database should reestablish the last solid state.

7. Temporal Logic in Information Systems

The Temporal Logic is used in the development of information system. It is used in the components of system related mainly to the general construction of programs, storage and query of data. In the outmost level, there is an interface man-machine, in the intermediate level the information systems, should include data processing mechanisms for the entrance, edition, analysis, visualization and exit of data finally in the innermost level there should be a system of database management.

The temporal logic is used in the elaboration of rules of integrity, restriction and queries related to temporal data depending on the model of data and in the elaboration of expressions of queries and implemented in language that corresponds to the employed model.

In order to obtain success in the query in the temporal database it is necessary that the languages of query can manipulate the embedded data [30].

8. Temporal Logic and Applications

The applications Temporal Logic in Computer Science are presented in the works of renowned researchers like C. Caleiro, Michel Fisher, Chiara Ghidini [27, 31] and others.

In the late seventies the modal style of temporal logic found an extensive new application in the area of computer science. It was applied and related to the specification and verification of programs, specially the simultaneous execution ones, that is, the processing is performed by two or more processors, which work in parallel, aiming at assuring the correct behavior of the programs where it is required to specify how the actions of the several processors are related [27]. The relative synchronism of actions should be coordinated with every care to assure that the integrity of information shared by the processors is maintained. Among the key notions, it is the difference among the properties of "liveness" of properties temporal logic, where Fp assures that the desirable states are obtained in the course of execution, and Gp that assures that the undesirable states are never reached.

Temporal Logic can be found also in applications of artificial intelligence, as the agents' construction, being one of those types agents BDI (Belief, Desire and Intention), that possess an architecture of belief, desires and intentions that need formalization so that they have real usefulness.

The formalization of such terms can be accomplished through logical representation. Usually, modal logic is used for those situations, but in the nineties, the logical formalism based on Temporal Logic ceased to be used, since allowing the analysis of future agents believe it can affect the desires and intentions of present [31].

Another application of Temporal Logic is in software engineering through the use of artificial intelligence techniques which use Temporal Logic to improve the environment of software development and quality. An example in the software engineering is the environment of PANDORA, a process machine based on the concepts of Temporal Logic which utilizes PROLOG language in its programming. This environment has an algorithm of application of rules which optimizes the steps of execution. In this case, the rules are implemented in Temporal Logic. This makes it possible to express and establish which activities are allowed each time, besides how these activities are synchronized.

The time is characterized by a simple sequential line of events. Temporal Logic emerges as a tool for the development of information systems since these systems are composed of hierarchically related modules. Temporal Logic is utilized in the components of the system related mainly in the construction of the programs as a whole, in the storage and data retrieval.

In the user's closest level, there is a man machine interface; in the intermediary level, the information system must have data processing mechanisms for the entry, edition, analysis, visualization, and output of data and in a more internal level in the system there must be a system to manage the database [32].

9. Conclusion

This work has presented the potentials forms of use of Temporal Logic in Information Systems, mainly in what it refers to as development. The Temporal Logic is used for the specification and verification of programs, mainly those that possess the characteristics of simultaneous execution. In the software engineering, the use of Temporal Logic is made through the use of techniques of IA, and one objective is the improvement of the construction environment and software quality.

The use of Temporal Logic in systems of information is emphasized mainly in the module of storage of data, as the database is the module that corresponds to the nucleus of the systems of information; since it is in this that, the temporal data is stored. Nowadays logic temporal applications are involved in providing solutions to problems encountered in the engineering, and medicine such as large database of genetic sequence.

The Temporal Logic can be seen as effective alternative in the treatment of data and the recovery of historical information. This is due to its main characteristic representing differing true values in instants different from time. The use of the Temporal Logic in information systems is of essential importance, mainly in those in that the evolution of data over a space of time is a factor to be considered in reliability and integrity of the database.

References

[1] Copeland, B. Jack. Arthur Prior. First published Mon Oct 7, 1996; substantive revision Sat Aug 18, 2007. http://plato.stanford.edu/entries/prior/. .

[2] MortariI, Cezar A.. Introdution of Logic. São Paulo: UNESP Publishing. 2001. p.60-65.

[3] Abbragnano, Nicola. Dictionary of Philosophy. 2º Edition. São Paulo: Mestre JOU Publishing. 1962. p.596-597. (in Portuguese).

[4] Vilela, Orlando. Logic Initiation. Dominus Publishing. São Paulo. 1964. p.109. (in Portuguese).

[5] Martins, H. G., Lambert-Torres, G., Pontin, L. F., “An Annotated Paraconsistent Logic”, COMMUNICAR Publishing, 2007 (in Portuguese).

[6] Gaines, B. R. Fuzzy Reasoning and the Logics of Uncertainty. Proceedings of the sixth international symposium on Multiple-valued logic Publisher: IEEE Computer Society Press. May 1976.

[7] Emerson, E. Allen. Temporal and Modal Logic. Chapter 16. Handbook of theoretical Computer Science. Edited by J. van Leeuwen. Elsevier Science Publishers B.V., 1990. p. 997-1067.

[8] Garson, James. Modal Logic. First published Tue Feb 29, 2000; substantive revision Sat May 5, 2007. http://plato.stanford.edu/entries/logic-modal/.

[9] Agotnes, Thomas; Hoek, Wiebe van der; Wooldridge, Michael. Normative System Games. AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. May 2007.

[10] Melo, Ivo Soares. Information Systems Administration. Thomson Learning Publishing. São Paulo. 2002. p. 21-35. (in Portuguese)

[11] Watson, Richard T.. Data Management: Data Base Organization. 3a. Edition. Publishing LTC. Rio de Janeiro. 2004. p. 29 - 40. (in Portuguese)

[12] Date, C. J.. Introduction Systems Data Base. 8ª. Edition Petropolis-RJ: Campus Publishing , 2004, 104 p. 621-658. (in Portuguese)

[13] Codd, E. F.. Data Models in Database Management. ACM SIGMOD Record, ACM SIGART Bulletin, ACM SIGPLAN Notices, Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling, Volume 11, 16 Issue 2 , 74 , 1.

[14] Chen, Peter. The Entity-Relationship Model: A basis for the Enterprise View of Data Proc. of National Computer Conference, 1977, AFIPS Press, Pages 77-84.

[15] Atkinson, M. P.; Bancilon, F.; Dewitt, D.; Maier, D.; Zdonik, S.. The Object-oriented database system manifesto. Proceedings of the first deductive and object-oriented database conference, Kyoto, 1989. p. 40-57.

[16] Stonebraker, Michael; Brown Paul. Object-Relacional DBMS: Tracking The Next Great Wave. Morgan Kaufman Publishers, Inc. Second Edition. 1999.

[17] Suciu, D.. An Overview of Semistructured Data, SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory). vol 29, 1998.

[18] Borges, karla Albuquerque Vasconcelos; Fonseca, Frederico Torres. Model of data geographic. http://ww.pbh.gov.br/prodabel/cde/publicacoes/1996/borges1996.pdf. (in portuguese)

[19] Navathe, Shamkant B.; Elmaris, Ramez. Data Base Systems. São Paulo: Pearson Addison Wesley, 4 ª Edition. 2006. p. 552-558.

[20] Korth, Henry F; Silberschatz, Abraham. Data Base Systems. Markon Brooks Publishing. 1995. 2ª Edition. p.17-22.

[21] COOD, E. F.. Relational Model of Data for Large Shared Data Banks. Communications of the ACM. Volume 13, No.6, June-1970, pg. 377 – 387.

[22] Simonetto, Eugênio de Oliveira. A Proposal for the Incorporation of Temporary Aspects, in the Logical Project of Databases, in SMDBs Relationa – PUCRS. Porto Alegre: Informatics of Institute Pontifícia Catolic University at Rio Grande do Sul, 1998. p.14.

[23] Jensen, Christian S.; Dyreson, Curtis; Böhlen, Michael; Clifford, James; Elmaris, Ramez; Gadia, Shashi K; Grandi, Fabio; Hayes, Pat; Jajodia, Sushil; Käfer Wolfgang; Kline, Nick; Lorentzos, Mitsopoulos, Yannis; Montanari, Angelo Montanari; Nonen, Daniel; Peresi, Elisa; Barbara, Pernici; Roddick , John F.; Sarda, Nandlal L.; Scalas, Maria Rita; Segev, Arie; Snodgrass, Richard T.; Soo, Mike D.; Tansel, Abdullah; Tiberio, Paolo; Wiederhold, Gio. The Consensus Glossary of Temporal Database Concepts - February 1998 Version. Proceedings of the 1998 ACM symposium on Applied Computing, p.235-240, February 27-March 01, 1998.

[24] Snodgrass, Richard; AHN, Ilsoo. A Taxonomy of Time in Databases. ACM SIGMOD Record Proceedings of the 1985 ACM SIGMOD international conference on Management of data SIGMOD '85, Volume 14 Issue 4. May 1985.

[25] Jensen, Christian S Temporal Database Management. 2000. <http://www.cs.auc.dk/~csj/Thesis/ >.

[26] Edelweiss, Nina. Temporal data Base: Theory and Pratice. Federal University at Rio Grande do Sul. Informatics of Institute. 1998.

[27] Caleiro, C.; Saake, G.; Sernadas, A.. Deriving Liviness Goals From Temporal Logic Specifications. Academic Press Limited. 1996.

[28] Chomicki, Jan; TOMAN, David; BÖHLEN, Michael. Querying ATSQL Databases with Temporal Logic. ACM Transaction on Database Systems, Vol.26, no. 2, june 2001.

[29] Chomicki, Jan; Toman, David. Temporal Logic in Information Systems. BRICS Research in Computer Science. Lecture Series- LS-97-1. ISSN1395-2048. November 1997.

[30] ATZENI, Paolo; CERI, Stefano; PARABOSCHI, Stefano. Database Systems: concepts, languages & architectures. McGraw Hill Publishing Company. 2000. p. 349-390.

[31] FISHER, Michael; Ghidini, Chiara. The ABC of Rational Agent Modelling. AAMAS'02, July 15-19, 2002, Bologna, Italy.

[32] Lago, P.; Manalti, G. Pandora: a temporal logic based process engine. Workshop in logic programming applied to software engineering. 1994.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-55

A Heuristic Approach to the Cable Routing Problem in Electrical Panels

Alexandre Erwin ITTNER [a] , Claudio Cesar de SÁ [b], and Fernando Deeke SASSE [c]

[a] *WEG Automação S.A., Departamento de Projetos, Engenharia e Automação, 89256-900 Jaraguá do Sul, SC, Brazil*
[b] *Universidade do Estado de Santa Catarina, Departamento de Ciência da Computação, UDESC, 89223-100 Joinville, SC, Brazil*
[c] *Universidade do Estado de Santa Catarina, Departamento de Matemática, UDESC, 89223-100 Joinville, SC, Brazil*

Abstract. In this paper, we present new results concerning the heuristic optimization of cable routing in electrical panels. The problem is modeled and a heuristic solution, using an insertion algorithm and a modified version of the Dijkstra's algorithm, is proposed, analyzed, and compared with human-made solutions. Tests have shown that good results can be obtained from layouts commonly found in the industry.

Keywords. cable routing, electrical systems design, optimization.

Introduction

In industrial facilities, like hydroelectric power plants, automobile manufacturing plants, and other facilities equipped with medium or large-sized industrial automation systems, there are panels and cabinets using dozens or hundreds of electrical components like contactors, relays, frequency inverters and PLCs. In major installations, these components are wired by thousands of cables, passing through hundreds of conduits arranged as a graph. The arrangement of these cables has direct influence over the cost of the installation and in the design quality. The optimal definition of the routes used by the cables also allows the definition of the quantity and type of the cables used in the panel during the design time.

The problem of giving a path for each cable in the panel, connecting all the associated components at a minimum cost and without overfilling the space available in the conduits, will be called here *the cable routing problem in electrical panels*. This is a real NP-complete problem from the industry, with a large scope, since many other subproblems are embedded in it.

The routing can be done manually: empirically or aided by measure systems (a ruler on a scaled drawing or a measurement tool in a CAD application), the designer selects the shortest path for every cable, starting from the most expensive and going to the cheaper ones. If the shortest route between two components becomes overfilled, the designer reroutes the cable through the shortest underfilled route available. If a feasible route cannot be found, the designer moves the already routed cables to another path. This process follows iteratively until all cables are routed.

This is a stressing, repetitive, and error-prone process. This paper analyzes the properties of cable laying in electrical panels and suggest a computer solution that near-optimal solutions for a simplified version of the problem.

1. Modeling

A panel is composed of an arbitrary number of components (contactors, overload relays, fuses, PLCs, etc.), each one with an arbitrary number of connection terminals, in a given physical position, and a set of conduits for wire disposition. Therefore, the model of a panel shows the connection terminals, the conduits and associations among them. Mathematically, the panel, P_n can be represented by

$$P_n := (L_t , L_c , L_i) \tag{1}$$

where L_t denotes the list of connection terminals, L_c denotes the list of conduits, and L_i denotes the list of connections. The designer gives the physical position of the components and terminals, according to design standards, and the routing process is not allowed to change it. The layout of a simplified panel is shown in Figure 1.

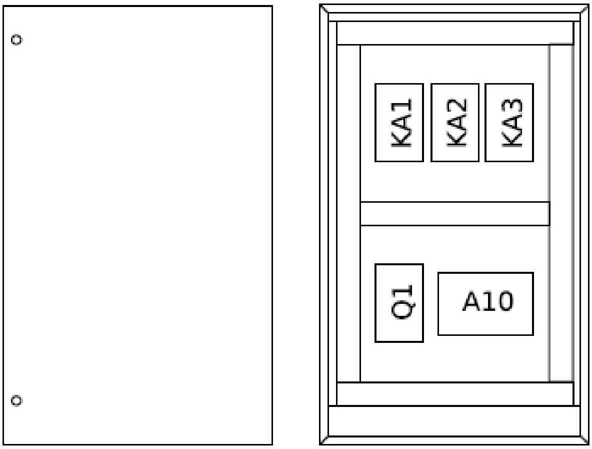


Figure 1 - . A typical panel

A connection I_n is a set of electrically equivalent terminals that must be connected by cables and is represented by

$$I_n := (L_t , e_i , s_e , c_n) \quad . (2)$$

where L_t denotes the list of connected terminals, e_i represents the electrical properties of the cable used for this connection (gauge, color, insulation voltage, maximum allowed temperature, etc.), s_e is the external cross-section of the cable, and c denotes the cost of the cable. The values of e_i are important regarding the electrical design, but are not used in the routing process. A connection among three terminals is represented in Figure 2.

Each connection gives origin to one or more cables, needed to electrically connect the terminals. A cable w is represented by

$$w_n := (t_i, t_f, e_i, s_e, c) \tag{3}$$

where t_i and t_f denote the starting and ending terminals, respectively, e_i represents the electrical properties of the cable, s_e its external cross-section, and c its cost.

Each cable from the same connection wires two terminals, and no terminal may be connected to more than two cables[1]. Therefore, $q_{term} - 1$ cables are needed to connect q_{term} terminals.

The connection order of the terminals from a connection list is particularly important because it allows some minimization on the distance traveled by the cables. From the combinatorial analysis, it can be shown that there are $q_{term}!$ permutations for the list of terminals and that half of these permutations are inversions of those already enumerated ones. As permutations with inverted connection order of the terminals are not electrically distinct, there are $q_{term}!/2$ ways to sort the terminals of each connection. This also means that the number of available sequences increases exponentially with the number of connected terminals.

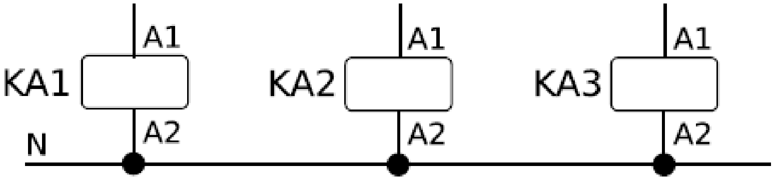


Figure 2 - Connection among three terminals of three distinct components

A terminal is represented by

$$T_n := (n_c, n_t, P_{xyz}) \quad . \tag{4}$$

where n_c is the component name (eg. KA1, KA2, and KA3 from Figure 2), n_t is the identification of the connected terminal (eg. A1 and A2), and P_{xyz} denotes the threedimensional point with coordinates (x, y, z) of the terminal within the panel space.

Conduits are line segments representing wire ducts, raceways, cable trays, and other materials used to hold cables in electrical panels. A conduit is represented by

$$C_n := (s_c, A_{xyz}, B_{xyz}, t) \tag{5}$$

where s_c denotes the cross-sectional area available for the cables, including safety margins, A_{xyz} and B_{xyz} respectively denotes the starting and ending points of the conduit in the panel space and t specifies the conduit type. The length of a conduit is given by the Euclidean distance between the points A_{xyz} and B_{xyz}. For modeling, there are two types of conduits:

[1] This is valid for terminals connected in a daisy chain. In other configurations, like those in distribution bars, more than two cables may exist.

Open conduits. Conduits that allow the crossing of cables through its walls, as open-slot wire ducts and some types of raceways. An example of this type of conduit is given with dashed lines in Figure 3. Open-type conduits may also be used to model "virtual conduits", as harnesses bound with cable ties.

Closed conduits. Conduits that do not allow crossing, as solid-slot wire ducts, internal raceways and liquid tight conduits. An example of this type of conduit is given with full lines in Figure 3.

The first and the last jump of each cable may pass through the walls of an open conduit if there is no ending nearer to it. The points where the cable crosses the conduit wall are called *entry points* (points *P1* and *P2* in Figure 3). These jumps must be considered in the calculation of the cable length and the conduit saturation. The entry points can be determined by solving the following problem: *given a line segment* $\overline{AB}$ *representing the conduit and a point C representing the terminal, find the entry point D over* $\overline{AB}$ *for that the length of the line segment* $\overline{CD}$ *is minimal.* This step is executed for each conduit, so, it will find the nearest entry point to the terminal. Strategies to deal successfully with these entry points are described in Section 5.

The conduit set may be represented as a weighted graph with edges connecting nodes attributed arbitrarily, preserving the topology of the panel, as shown in Figure 3. Conduits have a limited available internal space, so the amount of cables transiting through an edge is limited by the sum of their cross section areas. A conduit is called *overfilled* if it cannot hold more cables due to this limitation.

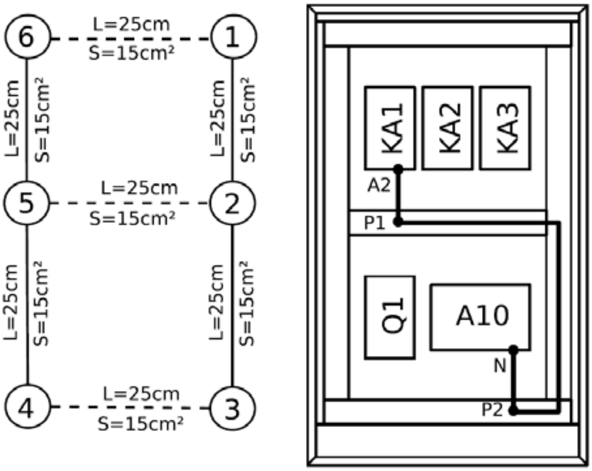


Figure 3 - Conduit graph of the panel from Figure 1 and routed cable

A route is, by definition, a sequence of nodes and terminals that gives the path followed by a cable from the starting terminal t_o and the ending terminal t_d, i. e.,

$$r_n := [t_o , n_1 , n_2 , \ldots , n_n , t_d] \tag{6}$$

where *n* are the nodes traveled by the cable. If the cable passes through *open* conduits, the entry points must be listed too.

The length *len(r_n)* of a route r_n is given by the sum of the lengths of the conduits traveled by the cable, including any entry point, i.e.,

$$len(r_n) := \sum_{i=1}^{i=m-1} dist(n_i, n_{i+1}) \tag{7}$$

where $dist(n_i,\ n_i{+}1)$ is the Euclidean distance between a n_i and n_{i+1}.

Formally, the *cable routing* problem is the problem of finding a set of cables $L_w = \{w_1, \ldots, w_m\}$ and a set of routes L_r for each connection i_n from the list of connections, so that the function

$$c_{total} := \sum_{j=1}^{j=n} c_{unit}(i_j) \times \sum_{k=1}^{k=m} len(R(w_k)) \tag{8}$$

is globally minimal. Here $c_{unit}(i_j)$ is the cost per unit of length of the cable used for the connection i_j , and *R(w)* is the function that links a cable *w* to a route from the set of all possible routes.

Given the set of conduits L_c of the panel and a set of cables L_w passing through a conduit $c \in L_c$, the function *R(w)* must satisfy the constraint

$$\sum sct(w) \leq s_c \forall w \in L_w, c \in L_c \tag{9}$$

where *sct(w)* is the external cross section area of the cable *w* and s_c the cross section area available in the conduit *c*.

2. Computational Complexity

The cable routing problem in electrical panels can be divided as a three-level optimization problem, since three major steps are needed for finding the optimal solution: (a) Given the connections among some terminals, generate a set of cables connecting them; (b) Route these cables through the shortest paths without violating the constraints in the conduits; (c) Sort these routes for minimal cost.

Routing one cable through the shortest path of a non-constrained graph is trivial and done in polynomial time with the Dijkstra algorithm. But this approach only finds the shortest path between two terminals and does not minimize the path of a set of cables needed to connect three or more terminals. Adding this requirement unfolds the problem into a combinatory optimization problem.

Adding the cost and constraints on conduit usage gives to the problem some similarity to the classical, NP-complete [1], *knapsack problem*: there are a bag of limited size (the set of conduits) and a set of objects (cables), with distinct costs, that must be inserted optimally in the available space. However, the cable routing problem has one more degree of complexity, since the cost of a cable changes as the problem evolves and the conduits become saturated.

3. PreviousWork

No work featuring exactly the same requirements as the problem described in Section 1 was found in the literature, but several problems with similar requirements were found.

Kloske and Smith [2] presented a solution to a cable routing and optimization problem using genetic algorithms. The problem described features cost minimization, raceway overfill, cable weight, and voltage drop. Unlike the cable routing problem in electrical panels, the problem presented does not have neither open conduits nor connections with more than two terminals. Ma *et al* [3] proposed a two-level genetic algorithm with two-level chromosome coding for a cable route optimization problem, combining route search and route combination into a hierarchical genetic algorithm.

The cable routing problem stated in Section 1 has several similarities with the problem of finding pathways in biochemical metabolic networks [4,5]. Dooms *et al* [6] introduced the graph-domain constraint programming (CP) and used this strategy to find that pathways. Viegas e Azevedo [7] repeated these results using constraint logic programming (CLP) with a new declarative, ECLiPSe-based, framework called GRASPER. These similarities and results suggest that graph-domain constraint logic programming may also be used to solve the cable routing problem in electrical panels.

4. Handling of Open Conduits and Partial Saturation

The presence of open conduits in panels raises some considerations regarding to the *partial saturation* of a conduit, that happens when some segment of a conduit becomes saturated, but other segments of the same conduit still having enough space for more cables. Figure 4 shows an example of a partially saturated conduit.

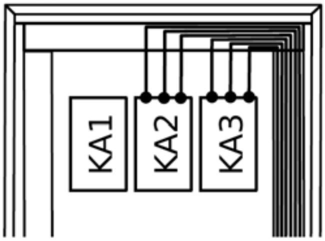


Figure 4 - Partial saturation of a conduit.

This problem may be addressed by adding a process called *conduit splitting* that transforms all open conduits in a set of closed conduits delimited by the entry-points of the cables crossing it. Several strategies may be devised for the conduit splitting, but three of them have practical significance:

a. Nearest entry point. The conduits are splitted exactly over the entry point, as shown in Figure 5a. This strategy is simple and gives the more precise results regarding to cable lengths, but may generate too many short edges, increasing the complexity of the graph and the solving time, without much improvements in the results.

b. Nearest entry point with approximations. This strategy avoids creating short edges by finding the entry point and searching for conduits ending nearer than a user-specified threshold distance t_d from that entry point. If there is such conduit, no splitting is performed and the cable enters the conduit by the existing ending, as shown in Figure 5b. This strategy gives good precision, but is slower than the former, since it also requires a search for conduit endings.

c. Constant length splitting. This strategy splits the conduits in a regular distance s_d, as show in Figure 5c. It is a fast strategy, since it does not need searches for existing edges nor nearest point calculations, but it may create useless conduit segments, i.e. conduit segments that will never be used because there is no near terminal.

Strategies *b* and *c* also allow the user to select the threshold distances and, therefore, control how much effort will be applied to the optimization of the routes. Threshold and splitting distances may also be heuristically selected according to the size of panel and properties of the circuit (i.e. shorter distances for small, heavily wired panels and longer distances for larger and sparsely wired ones).

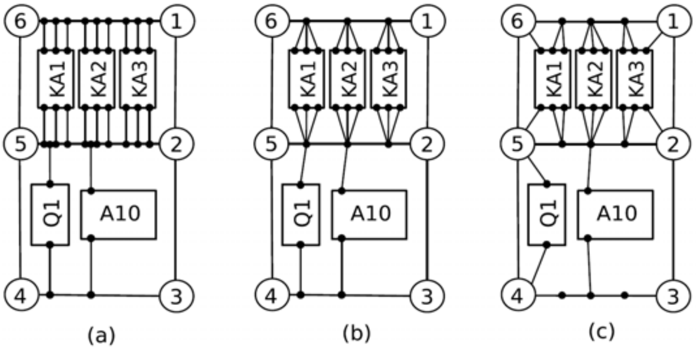


Figure 5 - Three strategies for conduit splitting: (a) nearest entry point, (b) nearest entry point with approximations, and (c) constant length splitting.

5. Simplification

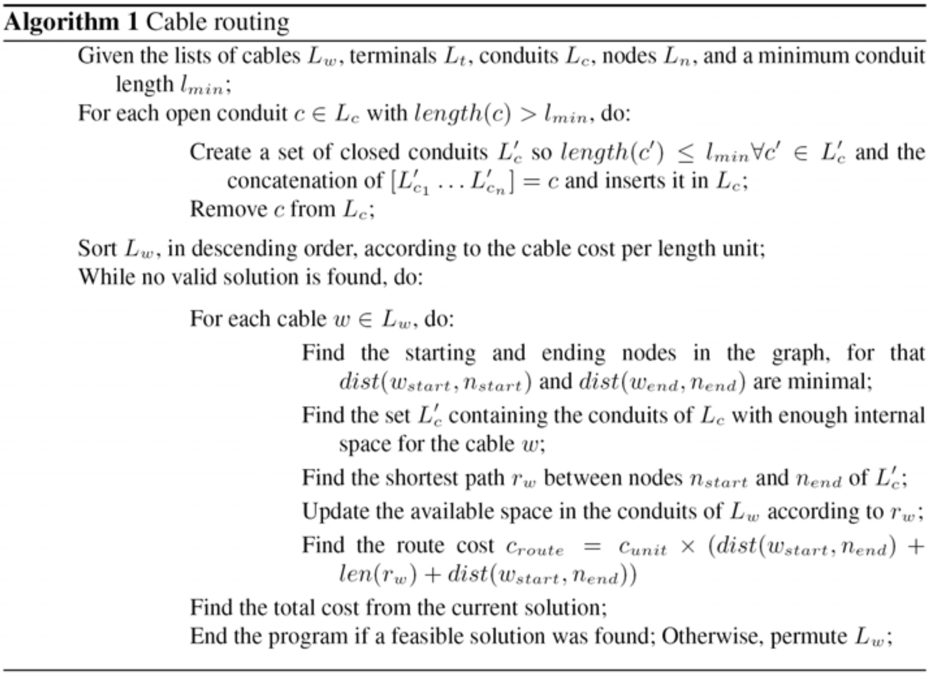
The model described in Section 1 allows high-quality solutions, but, its higher complexity inspires the search for a simplified model that allows fast near-optimal computer solutions. Therefore, the routing process was applied on a defined list of cables, not on a list of connections. This simplification cuts the computing complexity, but has the drawback of excluding the search for better solutions by changing the terminal wiring order. So, a cable will follow the shortest route between two terminals unless this route precludes, by conduit saturation, the existence of more economic routes for other cables.

6. Algorithm

The proposed algorithm (see Algorithm 1) runs in two steps: First, it performs the conduit splitting, using the “constant length splitting” strategy presented in Section 5 to convert all open conduits into closed ones. In the second step, it iterates through the list of cables, inserting them cables into the panel in descending order of cost, routing them through the shortest path, and decrementing the available space according to the cross-section area of the cable (this last process is called *graph shrinking*).

The cost minimization comes from the sorting, by inserting first the most expensive cables and giving them the shortest routes. The cheaper cables are routed later, getting increasingly worst routes due to conduit saturation.

The algorithm works with permutations to satisfy the saturation of cables in a specific conduit. This process is similar to the backtracking process available in languages as Prolog [8,9]. If, due to route saturation, a cable cannot be inserted, the current solution is discarded, the list of cables is permuted and the process begins again. If no permutation yields success, the problem is said to be non-feasible.



Algorithm 1 Cable routing

```
Given the lists of cables $L_w$, terminals $L_t$, conduits $L_c$, nodes $L_n$, and a minimum conduit
    length $l_{min}$;
For each open conduit $c \in L_c$ with $length(c) > l_{min}$, do:
        Create a set of closed conduits $L'_c$ so $length(c') \leq l_{min} \forall c' \in L'_c$ and the
            concatenation of $[L'_{c_1} \dots L'_{c_n}] = c$ and inserts it in $L_c$;
        Remove $c$ from $L_c$;
Sort $L_w$, in descending order, according to the cable cost per length unit;
While no valid solution is found, do:
        For each cable $w \in L_w$, do:
                Find the starting and ending nodes in the graph, for that
                    $dist(w_{start}, n_{start})$ and $dist(w_{end}, n_{end})$ are minimal;
                Find the set $L'_c$ containing the conduits of $L_c$ with enough internal
                    space for the cable $w$;
                Find the shortest path $r_w$ between nodes $n_{start}$ and $n_{end}$ of $L'_c$;
                Update the available space in the conduits of $L_w$ according to $r_w$;
                Find the route cost $c_{route} = c_{unit} \times (dist(w_{start}, n_{end}) +
                    len(r_w) + dist(w_{start}, n_{end}))$
        Find the total cost from the current solution;
        End the program if a feasible solution was found; Otherwise, permute $L_w$;
```

According to the insertion heuristics, it is expected that a solution can be found without the need of too many permutations. A solution is considered optimal if no cable was shifted from its shortest paths due to conduit saturation.

7. Implementation

The Algorithm 1 was implemented in the Lua programming language [10,11]. In this implementation, data files with information on the panel geometry and the wiring list are loaded by the application using the language's own parser, run through a validation routine, and used to build the adjacency matrix used by the variant of Dijkstra's Algorithm and tables of nodes, terminals and positions. Lua coroutines are used to generate permutations for the list of cables. An OpenGL viewer was also built to ease the validation of the models.

The implementation of the Dijkstra's Algorithm [12] used to find the shortest path between nodes differs from the standard algorithm by considering only conduits with enough space for the cables. Therefore, only the adjacency matrix is needed for each instance, lowering the need for processing. In order to allow the search on non-directed graphs, as the graph of conduits, the adjacency matrix keeps two references for each conduit.

7.1. Tests and Results

A case-study was performed comparing the solution generated by the implementation described in 7 with a solution given by a human expert, at the time of panel assembly, using his professional knowledge but without any formal procedure. It is the project[2] of a PLC remote panel with 57 segments of conduit and 692 wires and cables connecting 1096 distinct terminals.

The Figure 6 shows the physical layout of the components. This panel was chosen because there is a multitude of alternative routes and a high concentration of wires in the area near to the PLC, allowing to test the main characteristics of the algorithm. The data was originally generated from the wiring list used for the panel assembling and from a three-dimensional model, made with a CAD application, that gives the physical position of each terminal and conduit.

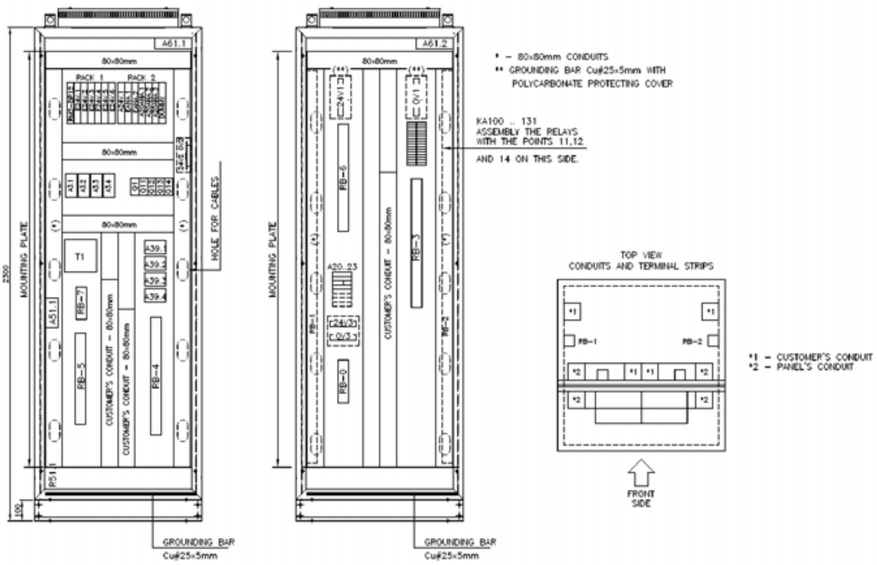


Figure 6 - Panel from the case study (Source: WEG Automação S.A.)

The test consists of running the program with the conduit splitting parameter set for: no splitting, 300mm, 200mm, 100mm, 50mm and 10mm. In all the tests, all cables are routed through the panel in the first iteration of the algorithm. These tests were performed in a Core 2 Duo 1.8GHz computer with 2GiB of RAM running Linux

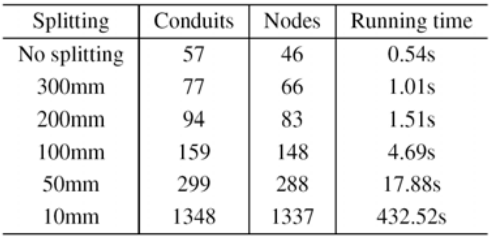
[2] Project number 035815E/05, October of 2005, courtesy of WEG Automação S.A.

and the LuaJIT 1.1.3 just-in-time compiler. To minimize timing errors, the program was run five times for each setting and the times presented are the average of these runs, measured with the Unix time command, and includes the time needed to start the interpreter, compile Lua code into machine code and load the data files.

Table 1 shows the number of nodes and conduits after the graph splitting and the corresponding running times. Table 2 shows the amount of cables calculated for each test and the amounts bought, at time of panel assembly, with assistance of the human expert using his professional knowledge, but without any formal procedure. Costs are given in Brazilian Reais.

This test shows that in all instances of shielded cable, the most expensive one, got good routes, causing substantial savings. The amount of the cheaper 0.75mm^2 dark blue cable given by the algorithm was higher than the amount given by the expert. Also, it must be noted that the expert rounded up the amounts that could not be safely calculated (the 4.0mm^2 green/yellow cable is an extreme case). The focus of the algorithm on the cost minimization may be inferred from the global cost decrease. The test also shows that lowering the minimum conduit length leads to more precise solutions, but also increases the running times exponentially.

Table 1 - Running times



Splitting	Conduits	Nodes	Running time
No splitting	57	46	0.54s
300mm	77	66	1.01s
200mm	94	83	1.51s
100mm	159	148	4.69s
50mm	299	288	17.88s
10mm	1348	1337	432.52s

8. Concluding Remarks and Future Work

The proposed algorithm gives good results when solving an instance of the problem that features the most common routing requirements, when the space available in the conduits is not tightly restricted. Solutions using the strategy for handling open conduits give more precise results than those using only closed conduits and it also allows the selection of the desired precision level by setting the minimum conduit length. As expected, the running times increases exponentially according to the selected precision level.

Future work involves a comparison between this approach with that one using Genetic Algorithms. It is worth mentioning that a comparison with [2] and [3] is not appropriate here, since in these papers the problems have no entry points. Another interesting development would be the application of graph-domain constraint logic programming in the generation of cable lists from the connection list. This would allow the implementation of the complete model described in Section 1.

Table 2. Cable usage comparison

Cable		Human expert		No splitting		300mm		200mm		100mm		50mm		10mm	
Gauge and color	(R$/m)	(m)	(R$)	(m)	(R$)	(m)	(R$)	(m)	(R$)	(m)	(R$)	(m)	(R$)	(m)	(R$)
0.5mm^2 two way shielded	1.69	100.00	69.34	69.34	117.18	79.67	134.65	76.69	129.60	77.86	131.58	77.22	130.50	77.41	130.82
0.75mm^2 dark blue	0.26	500.00	505.71	505.71	131.48	529.41	137.65	519.62	135.10	515.61	134.06	512.56	133.26	512.37	133.22
0.75mm^2 black	0.26	100.00	25.81	25.81	6.71	24.71	6.42	24.33	6.33	24.30	6.32	23.99	6.24	24.33	6.33
0.75mm^2 red	0.26	90.00	34.83	34.83	9.06	41.98	10.92	48.13	12.51	48.22	12.54	50.94	13.24	52.39	13.62
1.5mm^2 yellow	0.37	30.00	13.29	13.29	4.92	14.08	5.21	11.41	4.22	11.50	4.25	11.37	4.21	11.35	4.20
1.5mm^2 black	0.37	100.00	87.56	87.56	32.40	86.38	31.96	87.48	32.37	88.07	32.59	85.56	31.66	81.50	30.16
1.5mm^2 green/yellow	0.36	40.00	43.94	43.94	15.82	43.89	15.80	43.65	15.71	43.71	15.74	43.66	15.72	43.68	15.72
2.5mm^2 black	0.58	30.00	13.20	13.20	7.66	13.00	7.54	13.05	7.57	13.18	7.65	12.86	7.46	12.92	7.50
2.5mm^2 green/yellow	0.58	30.00	16.39	16.39	9.51	16.41	9.52	16.43	9.53	15.89	9.21	15.97	9.26	15.92	9.23
4.0mm^2 green/yellow	0.90	20.00	1.92	1.92	1.73	1.92	1.73	1.86	1.67	1.86	1.67	1.86	1.67	1.88	1.69
Total cost			463.70		336.47		361.40		354.61		355.61		353.22		352.49

References

[1] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice Hall, 1982. PAP ch 82:1 2.Ex.

[2] D. A. Kloske and R. E. Smith. Bulk cable routing using genetic algorithms. TCGA Report No. 94001, University of Alabama, Tuscaloosa, 1994.

[3] Xuan Ma, Kazuhiro Iida, Mengchun Xie, Junji Nishino, Tomohiro Odaka, and Hisakazu Ogura. A genetic algorithm for the optimization of cable routing. *Systems and Computers in Japan*, 37(7):61–71, 2006.

[4] Yves Deville, David Gilbert, Jacques van Helden, and Shoshana Wodak. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, 4(3):246–259, September 2003.

[5] Gregoire Dooms, Yves Deville, and Pierre Dupont. Constrained metabolic network analysis: discovering pathways using CP(Graph). In *Proceedings of the Workshop on Constraint Based Methods for Bioinformatics*, Sitges, Barcelona, Spain, October 2005.

[6] Gregoire Dooms, Yves Deville, and Pierre Dupont. Cp(graph): Introducing a graph computation domain in constraint programming. In *11th International Conference on Principles and Practice of Constraint Programming*, *Lecture Notes in Computer Science*, *No. 3709*, pages 211–225, Sitges, Barcelona, Spain, 2005. Springer-Verlag.

[7] Ruben Viegas and Francisco Azevedo. GRASPER: A Framework for Graph Constraint Satisfaction Problems. December 2007.

[8] Yoav Shoham. *Artificial Intelligence Techniques in Prolog*. Morgan Kaufmann Publishers, Inc, San Francisco, 5th edition, 1994. ISBN 1-55860-319-0 (cloth).

[9] L. Sterling and E. Shapiro. *The Art of Prolog*. MIT Press, Cambridge, Massachusetts, 2nd edition, 1994.

[10] Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes. The implementation of Lua 5.0. *Journal of Universal Computer Science*, 11(7):1159–1176, 2005.

[11] Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes. Lua: an extensible extension language. Software: *Practice & Experience*, 26(6):635–652, 1996.

[12] EdsgerW. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-67

Avatars Animation using Reinforcement Learning in 3D Distributed Dynamic Virtual Environments

Héctor Rafael OROZCO [a], Félix RAMOS [a], Jaime ZARAGOZA [a] and Daniel THALMANN [b]

[a] *Centro de Investigación y de Estudios Avanzados del I.P.N. Unidad Guadalajara*
Av. Científica 1145, Col. El Bajío, 45010 Zapopan, Jal., México
[b] *École Polytechnique Fédérale de Lausanne*
EPFL IC ISIM VRLAB Station 14 CH-1015 Lausanne, Switzerland

Abstract. The animation problem of avatars or virtual creatures using learning, involves research areas such as Robotics, Artificial Intelligence, Computer Sciences, Virtual Reality, among others. This work presents a Machine Learning approach using Reinforcement Learning and a Knowledge Base for animating avatars. This Knowledge Base (ontology) provides the avatar with semantic definition and necessary awareness of its internal structure (skeleton), its behavior (personality, emotions and moods), its learned skills, and also of the rules that govern its environment. In order to animate and control the behavior of these virtual creatures in 3D Distributed Dynamic Virtual Environments, we use Knowledge-Based Conscious and Affective Personified Emotional Agents as a type of logical agents, within the GeDA-3D Agent Architecture. We focus on the definition of minimum conscience of the avatars. The conscience and cognitive processes of the avatars allow them to solve the animation and behavior problems in a more natural way. An avatar needs to have minimum conscience for computing the autonomous animation. In our approach, the avatar uses the Knowledge Base first as a part of its conscience, and second to implement a set of algorithms that constitute its cognitive knowledge.

Keywords. Reinforcement Learning, Knowledge Base, Conscience, Avatar, Conscious Agent, GeDA-3D.

Introduction

The human mind has been studied for many philosophers for a long time. The human consciousness is considered to be one of the most interesting topics in the philosophy. This topic is called philosophy of mind. An important aspect of human consciousness is the self-knowledge or self-awareness, defined as the ability to perceive and reason about oneself. This aspect is highly developed in the Human being in comparison with other animals and it is considered very important for making agents with intelligent behavior. A Human being unaware of his or her personal characteristics, abilities and skills does not know what he or she can or cannot do, so he or she will have difficulties for interacting with others in a natural way. The *conscience* in general is defined as the knowledge that the Human being has of itself and of its environment.

Definition 1: The conscience of an avatar is the notion it has of its sensations, thoughts and feelings in a given moment within environment. That is, the avatar's conscience represents the understanding of its environment and its self-knowledge.

In this work, we use Knowledge-Based Conscious and Affective Personified Emotional (CAPE) Agents to develop the ability of avatar to perceive and reason about itself on the basis of the following: *Consciousness* (involve thoughts, sensations, perceptions, personality, moods and emotions), *stimuli and sensorial entrances* (relevant events), *introspection* (ability of avatar to reason about its perceptions and any conscious mental event), *awareness* (ability of avatar to be conscious; comprises perceptions and cognitive reactions to events, does not necessarily imply understanding), s*elf-consciousness* (awareness and understanding of avatar, it gives the avatar knowledge that it exists as a virtual entity separate from other avatars and virtual objects), and *qualia* (subjective properties of the perceptions and sensations of avatar).

In order to implement the conscience in an avatar, we use a Knowledge Base (KB). This KB represents the avatar conscience and it helps us to animate avatars or virtual creatures (VC). Thus, the KB (ontology) provides the semantic definition and necessary awareness of the internal structure of avatar (skeleton), its behavior (personality, emotions and moods), its learned skills, and also of the rules that govern its environment. We argue that consciousness is very important and plays a crucial role in creating intelligent agents with human abilities and skills. Thus, the avatar can be aware of how its skeleton is formed (considering its mobility and physical restrictions) and also of the rules that govern its environment.

The interest of this research is to give the avatars a basic conscience in order to have autonomous avatars able to act in 3D virtual environments. This means that it is not necessary to define previously the movements of the avatars to achieve a task in advance. Avatars must compute their movements themselves. But the generation of dynamic autonomous movements with high degree of realism is too complicated. Nevertheless, it is possible to make models of interactions between avatars and their environment in applications of computer animation and simulation [1]. For example, Virtual Humans (VH) can be used as virtual presenters, virtual guides, virtual actors or virtual teachers. Thus, the behavior and movements of VH can be controlled by using knowledge-based conscious emotional agents in order to show how humans behave in various situations [2]. There are different approaches to deal with the objective of this research. This research is a part of the GeDA-3D Agent Architecture [3, 4]. The following section is devoted to overview the related work.

1. Related Work

Interactive applications such as video games, collaborative virtual environments, simulations of virtual situations, films, among others, need believable virtual entities. But nowadays the behavior of these VC in current applications and systems is still very artificial and limited.

Definition 2: An avatar is a virtual entity with well-defined features and functionalities, able to live and interact in a 3D dynamic virtual environment.

Articulated models are often used for creating avatars. The animation of such models is often based on motion capture or procedurally generated motions. Despite the availability of such techniques, the manual design of postures and motions is still widespread; however, it is a laborious task because of the high number of degrees of freedom present in the models. Kinematic algorithms are also often used in the animation of avatars. Most of these algorithms require information about the position of joints, angles and limbs length. Next we will survey the most important related topics and give our opinion about them.

1.1. Motion Planning

Motion Planning (MP) has multiple applications. In Robotics it is used to endow robots of intelligence (autonomy), so they can plan theirs own movements. The problem of planning consists in finding a path for the robot from an initial point to a goal point without colliding with the obstacles in the environment [5]. In Artificial Intelligence (AI) the term *planning* takes a more interesting meaning. In this area the problems of planning are modeled with continuous spaces [6]. The problem of planning seems more natural and consists in defining a finite set of actions that can be applied to a discrete set of states and construct a solution by giving the appropriate sequence of actions. In [7] is presented a motion planner, which computes animations for virtual mannequins cooperating to move bulky objects in cluttered environments. In this work two kinds of mannequins were considered: human figures and mobile robot manipulators. Incremental Learning (IL) is a novel approach to the motion planning problem. It allows the virtual entities to learn incrementally on every planning query and effectively manage the learned roadmap as the process goes on [8]. This planner is based on previous work, on probabilistic roadmaps, and uses a data structure called Reconfigurable Random Forest (RRF), which extends the Rapidly Exploring Random Tree (RERT) structure proposed in the literature. The planner can take in account the environmental changes while keeping the size of the roadmap small. The planner removes invalid nodes from the roadmap as the obstacle configurations change. It also uses a tree-pruning algorithm to trim RRF into a more concise representation.

1.2. Motion Capture

In recent years the films have been successful exploding the technologies of motion capture. Motion capture is the process of capturing the live motion from a person or animal in order to animate an avatar [9]. Motion capture provides an impressive ability to replicate gestures, synthetic reproduction of large and complicated movements and behavior analysis, among others. At the moment the motion capture systems allow the collection of information for illustrating, studying and analyzing the characteristics of body limbs and joints during various motions, such as walking, running, etc. However, though impressive in the ability to replicate movement, the motion capture process is far away from perfect. Despite the longer time required to visualize the captured motion, the optical motion capture is often preferred to magnetic technology. The avatars animation design generates libraries of postures and motion sequences using a motion capture system and later combined the obtained data with standard editing tools. In the real-time motion generation, the avatars motions are based on the combination of pre-recorded sequences or dynamic motion captures, avoiding the recording stage.

1.3. Machine Learning

Machine Learning (ML) is a subfield of AI. This subfield covers the design and development of computer algorithms and techniques that improve automatically through experience. These algorithms allow the machines and intelligent agents to learn.

1.3.1. Reinforcement Learning

RL algorithms [10] allow machines and intelligent agents to automatically maximize their performance and learn their behavior, based on feedback from their environment within a specific context.

Definition 3: Reinforcement Learning is a subarea of ML and AI interested and involved in the problem of how an autonomous agent must learn to take optimal actions to achieve its goals in its environment, so as to maximize the notion of reward in long-term.

RL algorithms attempt to find a *policy* that maps *states* of environment to *actions* the agent ought to take in those states. The environment is typically represented using the *Finite Markov Decision Process* (FMDP). Each time the agent performs an action in its environment, a trainer may provide a reward or punishment (penalty) to indicate the desirability of the resulting state. Therefore, the task of agent is to learn from the reward indirectly. So, the agent can choose sequences of actions that produce the greatest cumulative reward.

1.3.2. Uses of Reinforcement Learning

Applications of RL are abundant. In fact, a lot of problems in AI can be mapped to a FMDP. This represents an advantage, since the same methodology can be applied to many problems with little effort. RL has been used in Robotics to control mobile robots and optimize operations in production lines or manufacture systems. An approach to animating humanoids was proposed in [11]. However, this approach has many restrictions in the used models. In [12] two well-known RL algorithms are presented (Q-Learning and TD-Learning). These algorithms are used for exploration, learning and visiting a virtual environment. In [13], RL algorithms are applied for the generation of Autonomous Intelligent Virtual Robots that can learn and enhance their task performance in assisting humans in housekeeping.

The potential for instructing animated agents through collaborative dialog in a simulated environment is described in [14]. In this work, STEVE, an embodied agent that teaches physical tasks to human students, shares activities with a human instructor by employing verbal and nonverbal communication. This way of work allows the agent to be taught in a natural way by the instructor. STEVE begins learning the task through a process of programming by demonstration. The human instructor tells STEVE how to observe his actions, and then it performs the task by manipulating objects in the simulated world. As the agent watches, it learns necessary information about the environment and procedural knowledge associated with the task.

A new way to simulate an Autonomous Agent's cognitive learning of a task for interactive virtual environment applications is proposed in [15]. This work is focused on the behavioral animation of virtual humans capable of acting independently. The concept of the Learning Unit Architecture that functions as a control unit of the Autonomous Virtual Agent's brain is proposed. The results are illustrated in a domain that requires effective coordination of behaviors, for example driving a car inside a virtual city. In [16], is presented an approach to integration of learning in agents for testing how it is possible to manage coherently a shared virtual environment populated with autonomous agents. Results proved that agents can automatically learn behavioral models to execute difficult tasks.

Learning Classifier Systems (LCS) are a ML paradigm introduced by John Holland in 1978. In LCS, an agent learns to perform a certain task by interacting with a partially unknown environment from which the agent receives feedback in the form of numerical reward. The incoming reward is exploited to guide the evolution of the agent's behavior which is represented by a set of rules, the classifiers. In particular, temporal difference learning is used to estimate the goodness of classifiers in terms of future reward; genetic algorithms are used to favor the reproduction and recombination of better classifiers [17]. In this work our approach is very different, because we use RL as a cognitive process that allows the avatar to learn new skills in its environment within a certain context. This difference is made by working with conscience that is not just knowledge but cognitive processes, which allow us to animate avatars in a more natural way. However, we are more interested in learning than in the exploitation of learning. That is, learning allows us to simulate the behavior and motion of life creatures into the avatars. This application constitutes a new use of RL. The following two sections are dedicated to the proposal of this research work. In these sections we will present the definition of an ontology proposed in order to define the internal skeletons of the avatars and the application of Reinforcement Learning (RL) for animating autonomously a nonhuman virtual arm.

2. Knowledge-Based CAPE Agents

Knowledge and reasoning are two essential elements for making intelligent agents able to achieve successful behaviors and take good actions in complex environments. These elements play a crucial role in dealing with partially observable environments. Knowledge-Based Agents are able to accept new tasks in form of goals. These agents can adapt to changes in the environment by updating its relevant knowledge about the environment and themselves.

Definition 4: CAPE Agents are Knowledge-Based Agents able to combine general knowledge with current perceptions to infer hidden aspects of the current state prior to taking new actions in their environment. Thus, these agents can increase their knowledge and learn new skills.

CAPE Agents represent a kind of logical agents whose knowledge always is defined. That is to say, each proposition is either true or false in the environment, although the agent may be unbeliever towards some propositions. Even though logic is a good tool to model CAPE Agents in partially observable environments, a considerable part of the reasoning carried out by agents depends on handling the uncertainty. However, logic cannot represent knowledge that is uncertain in a good way.

The main component of a knowledge-based agent is its KB. We use the KB as a set of logical sentences. Each logical sentence represents some assertion about the environment or the avatar. In order to add new sentences and query what is known to the KB, we use the standard sentences *TELL* (telling information to the KB) and *ASK* (asking information to the KB), respectively. Both sentences are used to generate new sentences basing on the old sentences. When we ask something to the KB, the answer depends on what we have told to it previously. In this work we propose a KB (ontology) to store and get knowledge about the internal structure of avatar (skeleton), its behavior (personality, emotions and moods) and its learned skills. The main objective is the exploitation and use of knowledge offered by the ontology in order to make autonomous animations of avatars using RL. This ontology allows sharing semantic information of avatars among CAPE Agents that live and interact in a 3D dynamic virtual environments created by a declarative description over the GeDA-3D Agent Architecture. In fact, the avatar uses the KB first as a part of its conscience and second to implement a set of algorithms that constitute its cognitive knowledge.

Figure 1 shows the relationships between the main classes of the proposed ontology. An avatar is defined using a morphology description (qualitative description) that defines its skeleton (geometry of avatar) and the anthropometry description (quantitative description) that offers information about its age, gender, weight and height. In addition, as a part of its behavior, an avatar has personality, emotions and moods. In this work we will only explain how we have defined the internal skeleton of avatar.

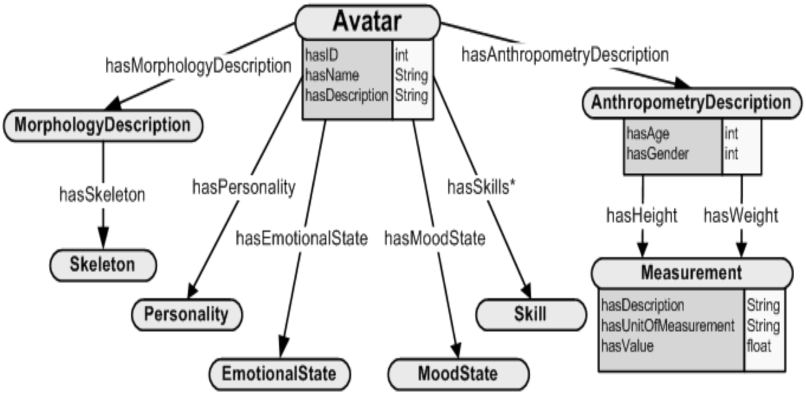


Figure 1. Semantic representation of avatar

The internal skeleton of avatar is composed by several parts. That is to say, bones and joints that form skeleton parts in specific (see figure 2). Each joint has a name and can have joints parents and/or joints children. There are motion constraints defined for each joint and a set of simple motions that define the alphabet of basic movements (micro-animation) that will be used to generate complex motions by means of combination between them (macro-animation). Also each bone can be united to one or more joints, and each joint has its position in the skeleton of avatar. Each bone of avatar has its measures that can be expressed in a predetermined unit of measurement, for example, in centimeters or decimeters. Using the previous definition and applying

RL algorithms, we can animate the internal skeleton of avatar as it will be shown in the following section.

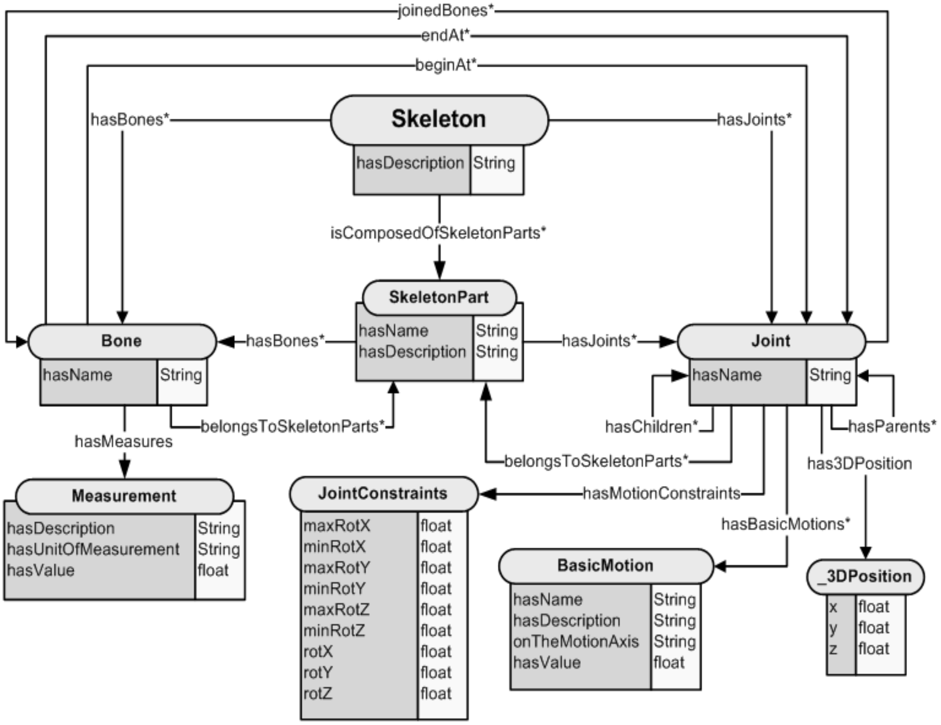


Figure 2. Skeleton definition of avatar

3. Avatars Animation using a Knowledge Base and Reinforcement Learning

In order to animate avatars it is necessary to use RL and MP algorithms to compute their motions. However in this work, we only present the use of RL. Avatars should be conscious of their internal structure (skeleton) and know how to combine simple movements to make complex activities or motions, which allow them to learn several skills and abilities. Using the previous definition and applying RL algorithms we can animate autonomous avatars. Figure 3 shows the proposed module of RL for the GeDA-3D Agent Architecture. The avatar should explore its body to know its structure and to learn a set of primitive motions, the basis for generating complex motions. In this work we propose the use of synergies (simple movements) to support the idea that the avatar cannot control all the degrees of freedom of its skeleton. For this reason a set of simple or primitive motions is selected (natural motions) to generate complex motions. Therefore, synergies are the base of the motions of avatar and can be manipulated by means of RL and MP algorithms.

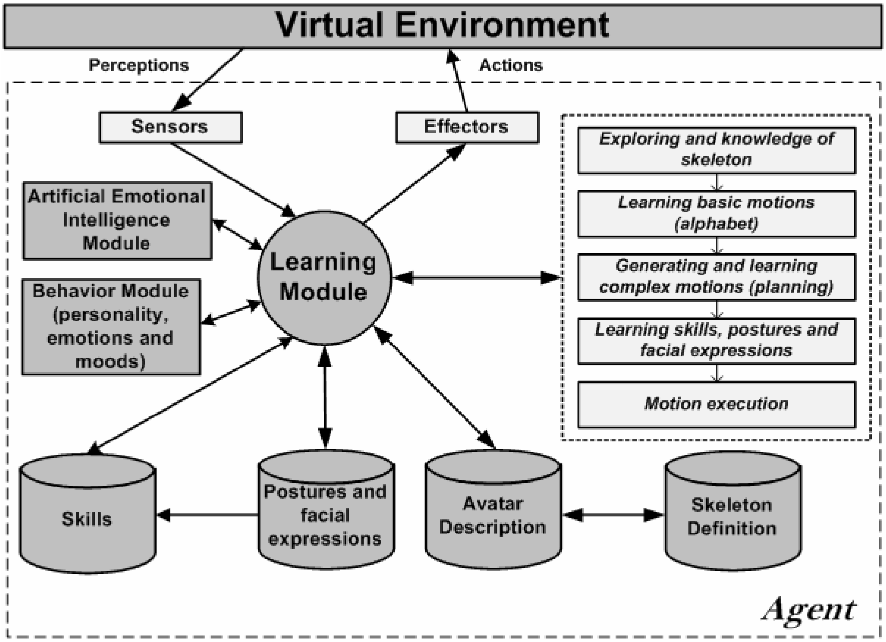

Virtual Environment
Perceptions
Actions
Sensors
Effectors
Artificial Emotional Intelligence Module
Behavior Module (personality, emotions and moods)
Learning Module
Exploring and knowledge of skeleton
Learning basic motions (alphabet)
Generating and learning complex motions (planning)
Learning skills, postures and facial expressions
Motion execution
Skills
Postures and facial expressions
Avatar Description
Skeleton Definition
Agent

Figure 3. Reinforcement Learning Module of the GeDa-3D Agent Architecture

3.1. Skeleton Parts involved in the Learning Task

We have to identify the used skeleton parts of avatar that we aim to animate on the basis of the following factors:

- Identifying the implied bones and joints of each skeleton part and their relations.
- Establishing the end-effectors and their locations in the skeleton parts.
- Defining a set of basic motions to each joint taking into account their motion constraints.

3.2. Learning Task Definition

It is very important to define clearly the learning task the avatar must perform. That is to say, we have to define the state and action sets used in the RL algorithm.

3.2.1. State Set

We define the state set $S = \prod J_i, \forall_{i=1,2,\cdots,n}$, where:

- State set of each joint J_i indicates each possible angle that can adopt each joint in order to accomplish its motion constraints, and,
- n indicates the number of joints in the skeleton.

3.2.2. Action Set

We translate the set of basic motions of each joint into the action set of RL task. Basic motions are to increase or decrease the angle related to such movement into small values, for example 5 or 10 degrees. Therefore, the action set A indicates the possible motions of each joint of the skeleton of avatar (degrees of freedom) over the three axis x, y and z. Next, we show the functions applied to find the action set and the state set used in the RL algorithm as part of cognitive knowledge of avatar:

```
//Action set of skeleton sk
function ACTIONS_SET (Skeleton sk, increment) returns the action set
begin
   Set A[]; //action set
   Joints joints[];
   //KB is the Knowledge Base
   joints = ASK(KB, JOINTS_SKELETON(sk));
   from i = 0 to i < joints.length do
        A = A + AXIS_ACTIONS(joints[i], x, increment)
              + AXIS_ACTIONS(joints[i], y, increment)
              + AXIS_ACTIONS(joints[i], z, increment);
   return A;
end

//Action set of joint j on the axis a
function AXIS_ACTIONS (Joint j, Axis a, increment) returns the action set
begin
   Set A[]; //action set
   vars min, max;
   //KB is the Knowledge Base
   min = ASK(KB, MINIMUM_ROT(j, a));
   max = ASK(KB, MAXIMUM_ROT(j, a));
   if max - min > 0 then
   begin
        A = new Set[2];
        A[1] = rotating the joint j in positive degrees on the axis a; //Increasing (in increment)
        A[2] = rotating the joint j in negative degrees on the axis a; //Decreasing (in increment)
   end
   return A;
end

//State set of skeleton sk
function ACTIONS_SET (Skeleton sk, increment) returns the state set
begin
   Set S[]; //state set
   Joints joints[];
   //KB is the Knowledge Base
   joints = ASK(KB, JOINTS_SKELETON(sk));
   from i = 0 to i < joints.length do
        S = S x JOINT_STATES(joints[i], increment);
   return S;
end

//State set of joint j
function JOINT_STATES (Joint j, increment) returns the state set
```

```
begin
   Set S[]; //state set
   S = S x AXIS_STATES (j, x, increment) x AXIS_STATES (j, y, increment)
           x AXIS_STATES (j, z, increment);
return S;
end

//State set of joint j on the axis a
function AXIS_STATES (Joint j, Axis a, increment) returns the state set
begin
   Set S[]; //state set
   vars max, min, aux, i;
   //KB is the Knowledge Base
   min = ASK(KB, MINIMUM_ROT(j, a));
   max = ASK(KB, MAXIMUM_ROT(j, a));
   aux = max - min / increment;
   if aux > 0 then
   begin
        S = new Set[aux];
        from i = 0 to aux do
          S[i] = min + i * increment degrees on the axis a;
        end
   return S;
end
```

3.3. Finite Markov Decision Process

In RL, an agent chooses the best action based on its current state. When this step is repeated many times, it turns into the problem that is known as the Markov Decision Process. Therefore, we considerer the representation of learning task to be the *Finite Markov Decision Process* (FMDP) based on the following statements:

- $P(s,a,s'): S \times A \times S \rightarrow [0,1]$ is the joint probability of making a transition to state s if action a is taken in the state s .
- $R(s,a,s'): S \times A \times S \rightarrow R$ is an immediate reward for making a transition from s to s' by action a .

Given any state and action, s and a, the probability of each possible following state s' is:

$$P^{a}_{ss'} = P\left\{s_{t+1} = s' \middle| s_t = s, a_t = a\right\}$$

Similarity, given any current state and action, s and a, together with any following state s', the expected value E of the following reward is:

$$R^{a}_{ss'} = E\left\{r_{t+1} \middle| s_t = s, a_t = a, s_{t+1} = s'\right\}$$

Basically the agent perceives a set of states S in its environment, a finite set of actions A that it can perform, and a set of obtained rewards in $\Re$. At each discrete time step t, the agent senses the current state $s_t \in S$ and the set of possible actions $A(s_t)$ for that state. When the agent takes an action $a \in A(s_t)$ and executes it, it receives the new state s_{t+1} and a reward or punishment r_{t+1} from the environment. Thus, the agent must learn to develop a policy $\pi : S \rightarrow A$, which maximizes the quantity $R = \sum_{t=0}^{n} r_t$ for a FMDP with a terminal state, or the quantity $R = \sum_{t=0}^{n} \gamma^t r_t$ for a FMDP without terminal states. Where $0 \leq \gamma \leq 1$ represents a discount factor used for future rewards. In fact, the agent does not necessarily know the reinforcement and next-state functions. These functions depend only on the current state and action. Before learning, the agent may not know what will happen when it chooses and executes a action in a particular state. The agent is only aware of its current state. This represents relevant information for the agent and allows it to decide which action to choose and execute. However, the policy is essential, because the agent uses its knowledge to choose an action in a given state.

3.4. Q-Learning Algorithm

There are several ways to implement the RL task. We have chosen the *Q-learning* algorithm, a well-known form of RL in which the agent learns to assign values to state-action pairs. In its simplest form, the *one-step Q-learning* algorithm is defined by the following *action-value function*:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left\lfloor r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \right\rfloor$$

We have fixed the values of α and γ to 0.5. The value γ is a discount factor used for future rewards. Thus, the agents can learn through experience. We have called a *state* each possible angle that can adopt each joint of the skeleton of avatar. Possible movements of each joint of the avatar's skeleton are called *action*. We can represent the above concepts using a state diagram. In this diagram a state is depicted by a node, while an action is represented by an arrow. We can put the state diagram and the instant reward values into a reward table or matrix R (reward function). The learning algorithm Q-learning is a simplification of RL. We need to put into the brain of agent a similar matrix named Q that will represent the memory of what the agent has learned through many experiences. The row of matrix Q represents current state of agent; the column of matrix Q pointing to the action indicates the following state. At the beginning, we have supposed the agent knows nothing, thus we put Q as a zero matrix. For simplicity in this work, we assume that the number of states is known. But in a better implementation, it is more convenient to start with a zero matrix of single cell and to add more columns and rows to the Q matrix if a new state is found. In general the transition rule of this Q-learning is as follows:

$$Q(state, action) \leftarrow R(state, action) + \gamma \max[Q(next\_state, actions)]$$

In the expression above the entry value in matrix Q (rows are states and columns are actions) is equal to corresponding entry of matrix R added by the multiplication of a learning parameter γ and maximum value of Q for all actions in the following state. Therefore, the agent will explore state after state until it reaches the goal. Each exploration is an *episode*. In one episode the agent will move from the initial state to the goal state. Once the agent has arrived at the goal state, the algorithm passes to the next episode. Each episode is equivalent to one training session. In each training session the agent explores the environment (represented by matrix R), gets the reward (or none) until it has reached the goal state. The purpose of the training is to enhance the brain of agent (represented by the matrix Q). More training will give better matrix Q that can be used by the agent to move in the most optimal way. Parameter γ has range of values form **0** to **1** ($0 \leq \gamma \leq 1$). If γ is closer to zero, the agent tends to consider only immediate reward. If γ is closer to one, the agent will consider that the future reward has greater weight and importance. That it to say, the agent will be willing to delay the reward. In order to use the matrix Q, the agent traces the sequence of states, from the initial state to the goal state.

3.5. Action Selection Rule

An important constraint of RL is the fact that only Q-values for actions that are tried in current states are updated. The agent learns nothing about actions that it does not try. The agent should try a range of actions in order to have an idea about what action is a good decision and what is not. That is to say, at any given moment of time, the agent must only choose an option: it can execute the action with the highest Q-value for the current state (*exploitation*), or it can execute an action randomly (*exploration*). Exploitation is based on what the agent knows about its environment, this probably can give more benefits to the agent. On the other hand, exploration offers the agent the possibility to learn actions that would not be tried otherwise. In order to deal with the exploration vs. exploitation dilemma, we choose an $\varepsilon - greedy$ method. First, we initialize ε to **1.0** and at the beginning of each episode we decrease it. Later, we update the value ε in a way inversely proportional to the number of elapsed episodes in the execution of the learning algorithm in the following way:

$$\varepsilon = 1.0 - \left(\frac{elapsed\_episodes}{total\_episodes} \right)$$

Therefore, the CAPE Agents learn using the following logical principles:

- If an action in a given state causes a bad decision, the agent learns not to execute that action in that situation.
- If an action in a given state causes a good decision, the agent learns to take that action in that situation.
- If all actions in a given state cause a bad decision, the agent learns to avoid that state. That is, the agent does not take actions in other states that would lead it to be in similar states.
- If any action in a given state causes a good decision, the agent learns to prefer that kind of states.

3.6. Reward Function

The *reward function* R is one of the most important components of RL because it defines the purpose or goal of agent in the learning task. We have to define or find a reward function that closely represents the agent's goals in the learning task. Our method consists in minimizing the *Euclidean distance* between $C_{x,y,z}$ (end-effector current position) and $G_{x,y,z}$ (end-effector goal position). That is to say, to minimize the total reward received in the long run. However, there exists a problem: the goal of RL is to maximize the total reward received in the long run, which is exactly the opposite of the RL goal. In order to fix this problem, we simply tag the Euclidean distance as a negative reward. Therefore, we define the reward function $R : S \times A \times S \rightarrow \Re^{-}$ as:

$$R(s,a,s') = -\sqrt{(c_x - g_x)^2 - (c_y - g_y)^2 - (c_z - g_z)^2}$$

Although we have defined correctly the reward function, $C_{x,y,z}$ is unknown and needs to be calculated from s'. That is to say, at each time step, we need to calculate the end-effector's current position $C_{x,y,z}$ given the agent's current state s'. This problem is known as the *Forward Kinematics Problem*. In order to solve this problem, we have used the **Denavit-Hartenberg convention** (D-H convention) [18] to select the frames attached to each joint of the skeleton of avatar in a systematic way. That is, we establish the joint coordinate frames using the D-H convention.

3.7. Results

Using the previous definition we animated a nonhuman virtual arm composed of two bones: *Humerus* and *Forearm* that are connected by two joints: *Shoulder* and *Elbow*. Figure 4 shows the first case study. In this case study we considered two degrees of freedom, one of them assigned to the shoulder and the other assigned to the elbow. Available movements for both shoulder and elbow are pitch (up or down). Possible actions that can be performed are to increase or decrease the angle of each joint (shoulder and/or elbow) by 10 degrees. We represent the state s as a 2-tuple (shoulderPitch, elbowPitch), where:

- shoulderPitch, elbowPitch $\in \{0°,10°,20°,\cdots,180°\}$.

Therefore the state set is:

$$S = \{0°,10°,20°,\cdots,180°\} \times \{0°,10°,20°,\cdots,180°\}$$
$$|S| = 19 \times 19 = 361$$

Angles of each joint are bounded in the rank from 0 to 180 degrees, this rank accomplishes with the motion constraints defined in the ontology (see figure 5). The action set represent all the available movements of the arm considered in this learning task. These motions are pitch up or down the shoulder ten degrees and pitch up or down the elbow ten degrees. Therefore, the action set is:

$$A = \begin{Bmatrix} shoulderPitchUP, shoulderPitchDown, \\ elbowPitchUp, elbowPitchDown \end{Bmatrix}$$
$$|A| = 4$$

In spite of the fact that the animation of this arm was made in a 3D environment, the movements it performs is in a 2D plane, due to the number of degrees of freedom the arm has.

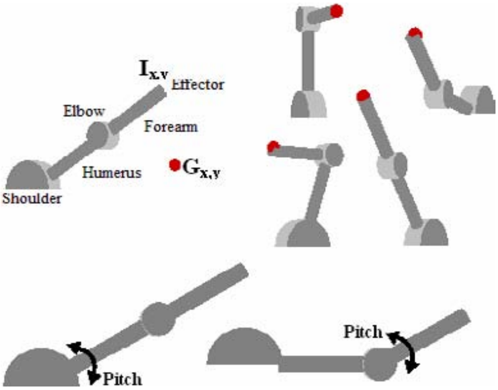
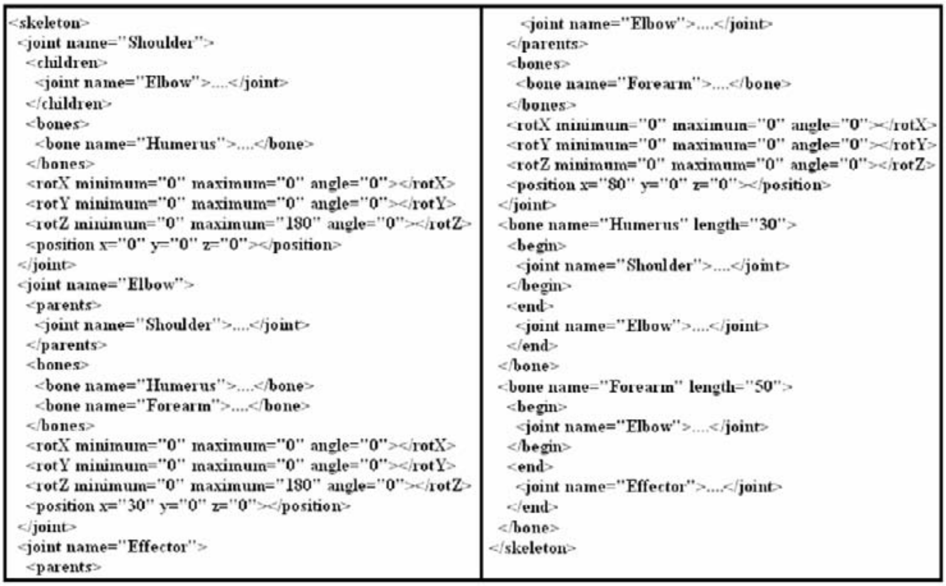


Figure 4. Reinforcement Learning in a nonhuman virtual arm with two degrees of freedom. The arm adopts a final configuration to collocate the end-effector in the goal position



```xml
<skeleton>
 <joint name="Shoulder">
  <children>
   <joint name="Elbow">....</joint>
  </children>
  <bones>
   <bone name="Humerus">....</bone>
  </bones>
  <rotX minimum="0" maximum="0" angle="0"></rotX>
  <rotY minimum="0" maximum="0" angle="0"></rotY>
  <rotZ minimum="0" maximum="180" angle="0"></rotZ>
  <position x="0" y="0" z="0"></position>
 </joint>
 <joint name="Elbow">
  <parents>
   <joint name="Shoulder">....</joint>
  </parents>
  <bones>
   <bone name="Humerus">....</bone>
   <bone name="Forearm">....</bone>
  </bones>
  <rotX minimum="0" maximum="0" angle="0"></rotX>
  <rotY minimum="0" maximum="0" angle="0"></rotY>
  <rotZ minimum="0" maximum="180" angle="0"></rotZ>
  <position x="30" y="0" z="0"></position>
 </joint>
 <joint name="Effector">
  <parents>
   <joint name="Elbow">....</joint>
  </parents>
  <bones>
   <bone name="Forearm">....</bone>
  </bones>
  <rotX minimum="0" maximum="0" angle="0"></rotX>
  <rotY minimum="0" maximum="0" angle="0"></rotY>
  <rotZ minimum="0" maximum="0" angle="0"></rotZ>
  <position x="80" y="0" z="0"></position>
 </joint>
 <bone name="Humerus" length="30">
  <begin>
   <joint name="Shoulder">....</joint>
  </begin>
  <end>
   <joint name="Elbow">....</joint>
  </end>
 </bone>
 <bone name="Forearm" length="50">
  <begin>
   <joint name="Elbow">....</joint>
  </begin>
  <end>
   <joint name="Effector">....</joint>
  </end>
 </bone>
</skeleton>
```

Figure 5. Definition of an nonhuman virtual arm with two degrees of freedom based on XML using the proposed ontology

Figure 6 shows the last case study. In this case study we animated other nonhuman virtual arm with four degrees of freedom (left arm). Three of them assigned to the shoulder and the last one assigned to the elbow. Available movements for this arm are: for the shoulder: roll (right, left), yaw (right, left) and pitch (up, down), and for the elbow: pitch (up, down). Possible actions that can be performed are to increase or decrease the angle of each joint (shoulder and/or elbow) in 10 degrees. In this case study, we have represented the state set S as a 4-tuple (shoulderRoll, shoulderYaw, shoulderPitch, elbowPitch), where:

- shoulderRoll $\in \{0°, 10°, 20°, \cdots, 90°\}$,
- shoulderYaw $\in \{0°, 10°, 20°, \cdots, 180°\}$,
- shoulderPitch $\in \{0°, 10°, 20°, \cdots, 240°\}$, and,
- elbowPitch $\in \{0°, 10°, 20°, \cdots, 150°\}$.

Therefore, the state set is:

$$S = \{0°, \cdots, 90°\} \times \{0°, \cdots, 180°\} \times \{0°, \cdots, 240°\} \times \{0°, \cdots, 150°\}$$
$$|S| = 10 \times 19 \times 25 \times 16 = 76000$$

Angles of each joint accomplishes with the motion constraints defined in the ontology (see figure 7). Possible motions are: roll the shoulder right or left by ten degrees, yaw the shoulder right or left by ten degrees, and pitch the shoulder and elbow up or down by ten degrees. Therefore, the action set is as follows:

$$A = \left\{ \begin{array}{l} shoulderRollRight, shoulderRollLeft, shoulderYawRight, \\ shoulderYawLeft, shoulderPitchUp, shoulderPitchDown, \\ elbowPitchUp, elbowPitchDown \end{array} \right\}$$

$$|A| = 8$$

In this case study the movements are performed in 3D. These movements are very similar to the movements performed by the left arm of a real Human being.

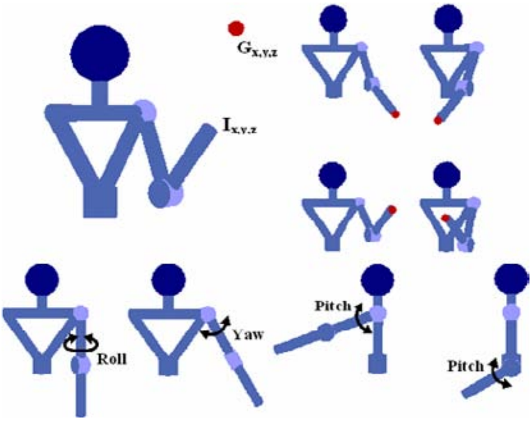
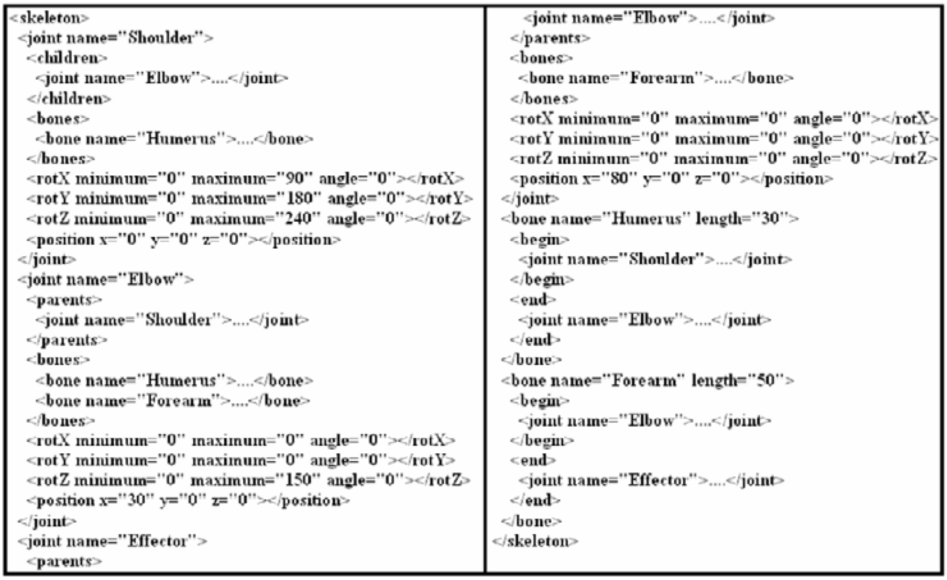


Figure 6. Reinforcement Learning in a nonhuman virtual left arm with four degrees of freedom. The arm adopts the final configuration to collocate the end-effector in the goal position



```
<skeleton>
 <joint name="Shoulder">
  <children>
   <joint name="Elbow">....</joint>
  </children>
  <bones>
   <bone name="Humerus">....</bone>
  </bones>
  <rotX minimum="0" maximum="90" angle="0"></rotX>
  <rotY minimum="0" maximum="180" angle="0"></rotY>
  <rotZ minimum="0" maximum="240" angle="0"></rotZ>
  <position x="0" y="0" z="0"></position>
 </joint>
 <joint name="Elbow">
  <parents>
   <joint name="Shoulder">....</joint>
  </parents>
  <bones>
   <bone name="Humerus">....</bone>
   <bone name="Forearm">....</bone>
  </bones>
  <rotX minimum="0" maximum="0" angle="0"></rotX>
  <rotY minimum="0" maximum="0" angle="0"></rotY>
  <rotZ minimum="0" maximum="150" angle="0"></rotZ>
  <position x="30" y="0" z="0"></position>
 </joint>
 <joint name="Effector">
  <parents>
    <joint name="Elbow">....</joint>
  </parents>
  <bones>
   <bone name="Forearm">....</bone>
  </bones>
  <rotX minimum="0" maximum="0" angle="0"></rotX>
  <rotY minimum="0" maximum="0" angle="0"></rotY>
  <rotZ minimum="0" maximum="0" angle="0"></rotZ>
  <position x="80" y="0" z="0"></position>
 </joint>
 <bone name="Humerus" length="30">
  <begin>
   <joint name="Shoulder">....</joint>
  </begin>
  <end>
   <joint name="Elbow">....</joint>
  </end>
 </bone>
 <bone name="Forearm" length="50">
  <begin>
   <joint name="Elbow">....</joint>
  </begin>
  <end>
   <joint name="Effector">....</joint>
  </end>
 </bone>
</skeleton>
```

Figure 7. Definition of a nonhuman virtual arm with four degrees of freedom based on XML using the proposed ontology

4. Conclusions

Nowadays, one of the great challenges in VR is to create avatars with characteristics proper to real living creatures. That is, it is desirable that these VC are able to reason, learn, feel and react as if they were intelligent creatures with cognitive capability to make decisions. The design of these VC has been a motivating task for researchers of different areas, such as Robotics, Virtual Reality (VR), AI and Computer Sciences (CS). The advance in such research areas is impressive, but there is still much work to be done. In the video game industry the users demand every day more sophisticated video games, in which they can enhance their presence in the virtual environment, navigate, perceive elements and interact with the VC. In the film industry is growing the interest to characterize actors in animated movies in a more realistic way. In order to attain this, we have proposed in this work a novel approach to the animation of avatars using RL. Our approach is based on the use of ML algorithms to provide the virtual creature with the capability of learning new skills. Although the presented methodology has proved to work well, its success depends enormously on how well we define the reward function. RL allows the CAPE Agents to learn their behavior on basis of feedback from the environment. This behavior can be learned only once, or adapting in the time. If the problem is modeled in a suitable way, RL algorithm can converge to the global optimum. This represents the ideal behavior to maximize the reward.

Intelligent agents can use the knowledge about their environment and themselves offered by the KB to make new inferences and to take good actions. This knowledge is represented by logical sentences and it is stored in the KB. In this work we have used knowledge-based agents composed of a KB and an inference mechanism. These agents store logical sentences about the world and themselves in the KB, using the inference mechanism to infer new sentences (new knowledge). Agents use these sentences in order to decide which actions to take in a given moment. In this work, we have presented the necessary bases to implement the conscience of an avatar, shown how to define the internal skeletons of avatars, and described some of the main issues to be solved. The current version of our ontology is a work in progress. In this work we have shown some of the main possibilities of use of the ontology in order to animate articulated VC. Our research is focused on advancing the development of the ontology proposed. Our ontology plays a fundamental role in the animation of articulated virtual creatures controlled by conscious and intelligent agents. Knowledge Bases also have very important potential in the motion planning and motion learning in avatars. Our future work includes the generalization of the avatar animation with the use of Knowledge Bases, MP and RL techniques. These results will be applied to the development of semi-autonomous and autonomous avatars that interact in 3D distributed dynamic virtual environments over the GeDA-3D Agent Architecture.

References

[1] H. Schmidl and M. Lin. Geometry-Driven Physical Interaction Between Avatars and Virtual Environments. Computer Animation and Virtual Worlds, Vol. 15, non. 3-4, pages 229-236, 2004.

[2] S. Göbel, A. Feix, and A. Rettig, Virtual Human: Storytelling and Computer Graphics for a Virtual Human Platform. International Conference on Cyber Worlds, 2004.

[3] F. Zúñiga, F. F. Ramos and I. Piza. GeDA-3D Agent Architecture. Proceedings of the 11th International Conference on Parallel and Distributed Systems, pages 201–205, Fukuoka, Japan, 2005.

[4] H. I. Piza, F. Zúñiga and F. F. Ramos. A Platform to Design and Run Dynamic Virtual Environments. Proceedings of the 2004 International Conference on Cyberworlds, pp. 78-85, 2004.

[5] F. Schwarzer, M. Saha and J. Latombe. Adaptive Dynamic Collision Checking for Single and Multiple Articulated Robots in Complex Environments. IEEE Transactions On Robotics, Vol. 21, no. 3, pages 338-353, June 2005.

[6] S. M. LaValle. Planning Algorithms, Cambridge University Press, 2006.

[7] C. Esteves, G. Arechavaleta and J. P. Motion Planning for Human-Robot Interaction in Manipulation Task. IEEE International Conference on Mechatronics and Automation, Vol. 4, pages 1766- 1771 Laumond, 2005.

[8] T.-Y. Li and Y.-C. Shie. An Incremental Learning Approach to Motion Planning with Roadmap Management. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, pages 3411–3416, 2002.

[9] L. Herda, P. Fua and D. Thalmann. Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion. Computer Animation, pages 77-83, Philadelphia, PA, USA, 2000.

[10] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

[11] J. Peters, S. Vijayakumar, and S. Schaal, Reinforcement Learning for Humanoid Robotics. International Conference on Humanoid Robots, pages 1-20, Karlsruhe, Germany, September 2003.

[12] T. CondeW. Tambellini and D. Thalmann, Behavioral Animation of Autonomous Virtual Agents Helped by Reinforcement Learning. Lecture Notes in Computer Science, vol. 272, pages 175-180, Springer-Verlag: Berlin, 2003.

[13] T.-Y. Li and Y.-C. Shie. An Incremental Learning Approach to Motion Planning with Roadmap Management. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, pages 3411–3416, 2002.

[14] A. Scholer, R. Angros Jr., J. Rickel, and W. L. Johnson. Teaching Animated Agents in Virtual Worlds. In Smart Graphics: Papers from 2000 AAAI Spring Symposium, pages 46–52, 2000.

[15] T. Conde and D. Thalmann. Autonomous Virtual Agents Learning a Cognitive Model and Evolving. In Proceedings of the 5th International Working Conference on Intelligent Virtual Agents, IVA 2005, pages 88–98, 2005.

[16] T. Conde and D. Thalmann. Learnable Behavioural Model for Autonomous Virtual Agents: Low-Level Learning. In Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, pages 89–96, 2006.

[17] J. H. Holland, L. B. Booker, M. Colombetti, M. Dorigo, D. E. Goldberg, S. Forrest, R. L. Riolo, R. E. Smith, P. L. Lanzi, W. Stolzmann, and S. W. Wilson. What is a Learning Classifier System? In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, Learning Classifier Systems. From Foundations to Applications, volume 1813 of LNAI, pages 3–32, Springer-Verlag: Berlin, 2000.

[18] M. W. Spong and M. Vidyasagar. Robot Dynamics and Control. John Wiley & Sons, Inc., 1989.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-85

Annotated Paraconsistent Logic

Helga Gonzaga MARTINS [a], Carlos Henrique Valério de MORAES [a], Cláudio Inácio de Almeida COSTA [a], Germano LAMBERT-TORRES [a] and Antônio Faria NETO [b]
[a] *Itajuba Federal University*
Av. BPS 1303 – Itajubá – 37500-903 – MG – Brazil
[b] *University of Taubate*
Visconde do Rio Branco, 210 – Taubate – 12020-040 – SP – Brazil

Abstract - This paper presents a general view from the Two-Valued Annotated Paraconsistent Logic - 2vAPL to the Four-Valued Annotated Paraconsistent Logic- 4vAPL. The purpose to expand 2vAPL to 4vAPL is to enable the insertion of opinions from Experts in the knowledge base, so that the problems described approach their real condition, once the opinion of an Expert may be decisive in the evaluation of a system. Furthermore, with the expansion of 2vAPL to 4vAPL, it is possible to analyze the behavioral evolution of the opinions from these Experts within time.

Keywords. Paraconsistent Logic, Annotated Paraconsistent Logic, Non-Classic Logic.

Introduction

Many times, real situations do not fit the analysis of classical logic, which is limited when dealing with situations of inconsistencies, lack of definition, ambiguities, etc. Consequently, more efficient logical systems are needed to manipulate directly all this range of information that describes the real world.

Paraconsistent Logic constitutes a knowledge which was developed due to the necessities imposed by everyday life and guided by intuition. The results of the studies presented in this work reveal that the application methods of Paraconsistent Logics are alternative methods to give adequate treatment to these situations.

Inconsistent, indefinite, and partial knowledge situations naturally emerge from the description of the real world; despite this, men can reason adequately. However, this reasoning is not dealt under the view of Aristotelian logic, that is, that view in which any statement about something is either true or false. For instance, in control systems, the theory base is classical logic. Due to their binary structure (true or false), reasoning must be either simplified, disregarding facts or situations of inconsistencies or even vaguely summarized. This is so because in the complete description of the real world, time becomes considerably long when one deals with only two states [1].

Since real situations do not completely fit the binary forms of classical logic, several researchers have struggled to find other ways which better frame other concepts like lack of definition, ambiguities and inconsistencies, thus non-classical logic arises.

This research methodology consists of designing interpretation methods of Paraconsistent Logic considering its theoretical structure presented in previous relevant researches, as in [2], [3], [4], [5] and [6]. Application methods will be developed from these interpretations. These methods will carry out a treatment of the uncertain knowledge translating them into practical and theoretical concepts.

1. Historical Aspects

Paraconsistent Logics have had two forunners, A. N. Vasil'év and J. Lukasiewicz, who in 1910, in Russia and Poland respectively, discussed the possibility of a Paraconsistent Logic. However, their work was very limited because they referred to only changes that could be introduced in Aristotelian logic in case there was a non-validation of the Principle of Non-Contradiction.

Nonetheless, the first logician to formulate a Paraconsistent Propositional Calculus within the rigor of modern rules was the Polish logician S. Jáskowski in 1948. Jáskowski worked in his logic of paraconsistent nature called "Discursive Logic" under the influence of Lukasiewicz. However, he limited to development of a Discursive Propositional Calculus without extending it to superior order logic, which without much rigor, is the one that has several quantifiers so as to manipulate equality.

Newton C. A. da Costa was the first logician in Brazil in 1954 to develop Propositional Calculus, Quantificational Calculus with or without Equality, Theories of Descriptions, and Theories of Paraconsistent Set. Rigorously, a logical system is only accepted when there is a superior order calculus. It may be said that Paraconsistent Logic was created by Newton C. A. da Costa.

Jáskowski's logic itself was only developed in the 1960s when Da Costa and the Polish disciples of Jáskowski: L. Dubikajtis and J. Kotas amplified Discursive Propositional Calculus, built Discursive Quantificational Calculus of Superior Order and Discursive Set Theory. Moreover, Da Costa and Kotas extended Jáskowski's basic idea, which defined Discursive Logic to any unary operator, in any system.

Following these initial studies, Paraconsistent Logics have evoluted and presented great contributions to technological applications. For many reasons, Paraconsistent Logics have been converted into a vast research field in several centers around the world.

2. Philosophical Basis of Paraconsistent Logics

Paraconsistent Logics may be considered as those that do not validate the Law of Non-Contradiction. Evidently, if a contradiction is accepted under the point of view of these logics, then the classical rule is invalid. That is, the statement that any formulable proposition is inferred from a contradiction is invalid.

For Lukasiewicz, Aristotle himself, predecessor of classical logic, suggested the possibility of invalidating the Principle of Non-Contradiction. This would imply the supposition, though implicitly, the existence of Paraconsistent Logics in his work. It does not seem, however, as Lukasiewicz suggests, that Aristotle could be considered the creator of Paraconsistent Logics, since no implicit formalization of such logics can be found in his work. Nonetheless, he might be considered, in the history of Paraconsistent Logics as one of its most important predecessor.

A conceptual distinction between "Paraconsistency" and "Paraconsistent Logics" must be made. The first term, as has been used, involves everything that refers to contradiction and the corresponding law, as for example, Heraclitus' idea, the

Dialectic, the study of paradoxes, and the mentioned logics to deal with contradictory theories. Paraconsistent Logics, on the other hand, constitute a technical science founded on methods analog to those used by the standard logic. Paraconsistency is much wider than Paraconsistent Logics, involving technical as well as philosophical matters as in [7] and [8].

Moreover, Paraconsistent Logics show, as other Non-Classical Logics, that it is not licit nowadays to talk of a universal rationality, but as can be seen from scientific development, one should ask about what rationality is being dealt.

As science evolutes, Paraconsistent Logics give formal support for the existence of other theories concerning the real contradictions. Regarding this, it is convenient to cite Da Costa when he says that "… we call contradiction or inconsistency any pair of proposition, one is the negation of the other. Paraconsistent Logic does not exclude the possibility that both propositions of a contradiction are True. It does not exclude the existence of real true contradictions, that is, contradictions whose components refer to the real world. We think that we can eliminate this possibility. To know whether there are true contradictions in the real world constitutes an empirical issue; it is only decidable by the Empirical Sciences" [3].

3. Initial Considerations on Annotated Paraconsistent Logic

In this work, the lattice associated to Two-valued Annotated Paraconsistent Logic - 2vAPL will be studied according to references [9], [10] and [11]. A detailed analysis was performed on the Degrees of Belief and Disbelief values when binary, ternary, discreet in N points (multivalued), and continuous in time (analogic). This analysis was performed with the use of the Unitary Square of the Cartesian Plan- USCP.

In this study, the concept of Unitary Square of the Cartesian Plan- USCP is used as a basic tool to represent the new interpretation proposal of the new notable points and the logic operators for Two-valued Annotated Paraconsistent Logic - 2vAPL.

In the following section, some very important aspects of the USCP are considered for the practical applications of the results.

4. New Interpretation of the 2vAPL

For the Two-valued Annotated Paraconsistent Logic - 2vAPL, an interpretation I is a function I: P → $|\tau|$. A valuation VI: F → [0,1] X [0,1] is associated to each interpretation I, where F is the set of all the formulas. This can be seen graphically in the USCP.

The USCP presents values of x and y varying in a real closed interval [0,1], in such a way that these values represent the Degrees of Belief, μ_1 and of Disbelief, μ_2 respectively.

The points marked in Figure 1 determine the notable points in the USCP.

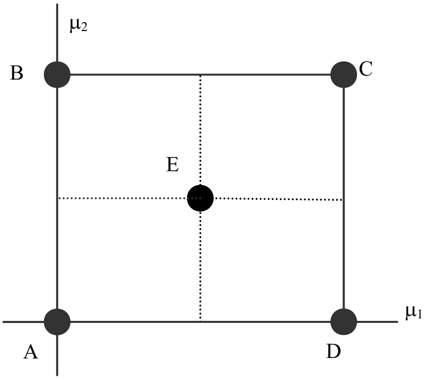


Figure 1 - Notable points in the USCP.

Point A =	(0, 0)	⇒	Paracomplete	(⊥)
Point B =	(0, 1)	⇒	False	(F)
Point C =	(1, 1)	⇒	Inconsistent	(I)
Point D =	(1, 0)	⇒	True	(T)
Point E =	(0.5; 0.5)⇒		Undefined	(U)

Thus, one may see that point E, referring to the USCP, is average value of the segment $\overline{FT}$ in Figure 2, equivalent to the unknown or undefined value (I) in the three-valued logic [12].

In the USCP, the segment $\overline{FT}$ is named perfectly consistent line (PCL) and the segment $\overline{\perp I}$ is named perfectly inconsistent line (PIL), according to Figure 2.

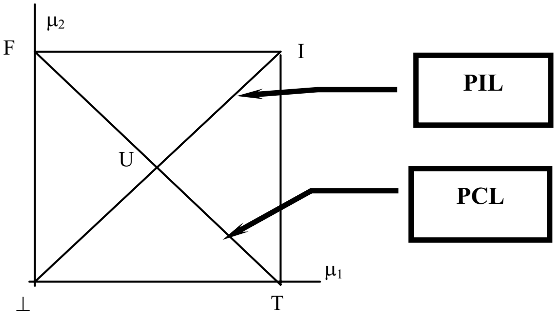


Figure 2 - Unitary Square and the Perfectly Consistent and Inconsistent Lines (PCL and PIL).

On the PCL, for any value of the Degree of Belief, the corresponding value of the Degree of Disbelief is its complement in relation to the unity. Therefore:

$$\mu_1 + \mu_2 - 1 = 0 \qquad (1)$$

When the meeting point between the Degree of Belief and of Disbelief is located above the PCL, it is considered a perfectly consistent point. Even though it represents partial or incomplete knowledge they present coherence between the Degrees of Belief and Disbelief. Therefore:

$$\mu_1 + \mu_2 = 1 \qquad (2)$$

Given a pair (μ1,μ2), the Degree of Uncertainty (DU) of this pair is the distance of the parallel straight line, which contains the pair, to the PCL. The DU receives a positive signal (+) if it is above the PCL and negative (-) otherwise. Thus, the perfectly consistent points have DU equal to zero.

The equation that defines the Degree of Uncertainty is:

$$DU = \mu_1 + \mu_2 - 1 \qquad (3)$$

It can be seen that as the point corresponding to the ordered pair (μ_1,μ_2) in the USCP gets away from the PCL, toward the ordered pair (1,1), there is a gradual increase of the DU up to its maximum value 1, located at point I.

In the same way, as the point corresponding to the ordered pair (μ_1,μ_2) in the USCP gets away from the PCL, toward the ordered pair (0,0), there is a gradual increase of the DU module up to its maximum value 1, located at point ⊥.

In relation to the PIL, one may observe that for a certain value of the Degree of Belief there corresponds an equal value of the Degree of Disbelief. The expression may be written as follows:

$$\mu_1 - \mu_2 = 0 \qquad (4)$$

When the meeting point between the Degree of Belief and of Disbelief is located above the PIL, it is considered a perfectly inconsistent point, such that it presents maximum inconsistency.

Therefore:

$$\mu_1 = \mu_2 \qquad (5)$$

Then, the DC on this line is zero.

Just like the DU, the mathematical equation that expresses the involvement of the Degrees of Belief, Disbelief, and of Certainty is as follows:

$$DC = \mu_1 - \mu_2 \qquad (6)$$

A DC different from zero defines a point pertaining to a straight line parallel to the PIL. It can be seen that as the point corresponding to the ordered pair (μ_1,μ_2) of the USCP gets away from the PIL, toward the ordered pair (1,0), there is a gradual increase of the DC until it gets to its maximum value of 1, located at point T.

Analogically, as the point corresponding to the ordered pair (μ_1,μ_2) of the USCP gets away from the PIL, toward the ordered pair (0,1), there is a gradual increase of the DC module until it gets to its maximum valor of 1, located at point F.

Thus, for each ordered pair composed by the value of the Degree of Belief μ_1 and by the value of the Degree of Disbelief μ_2, one may find the values of the Degrees of Uncertainty and of Certainty according to the equations below. They are represented in the USCP, according to Figure 3:

$$DU = \mu_1 + \mu_2 - 1 \qquad (7)$$

$$DC = \mu_1 - \mu_2 \qquad (8)$$

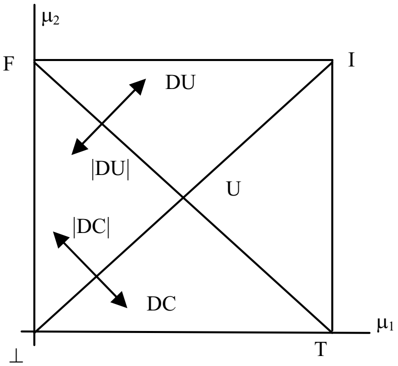


Figure 3 - USCP and the representation of the increase of the Degree of Certainty and Uncertainty.

The affirmations below may be made:

a - "If the Degree of Uncertainty is positive, DU > 0, then the resulting point will be above the perfectly consistent line, PCL".

b - "If the Degree of Certainty is negative, DC < 0, then the resulting point will be above the perfectly inconsistent line, PIL".

The joining of these two conditions delimits the region presented in Figure 4 below:

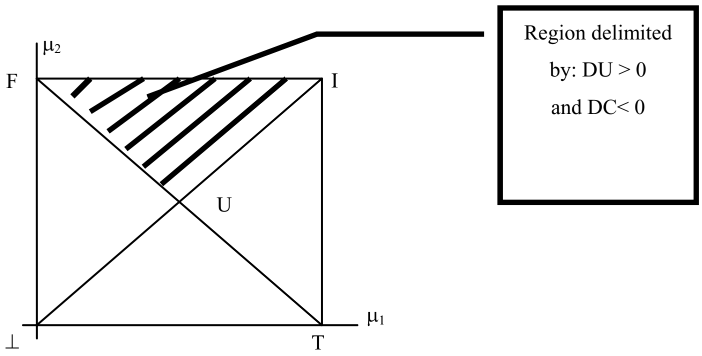


Figure 4 – Unitary Square highlighting the union of regions DU > 0 and DC < 0.

Two more affirmations can be made:

c - "If the Degree of Uncertainty is positive, DU > 0, then the resulting point will be above the perfectly consistent line, PCL".

d - "If the Degree of Certainty is positive, DC > 0, then the resulting point will be above the perfectly inconsistent line, PIL".

These two conditions delimit the region represented in Figure 5 below:

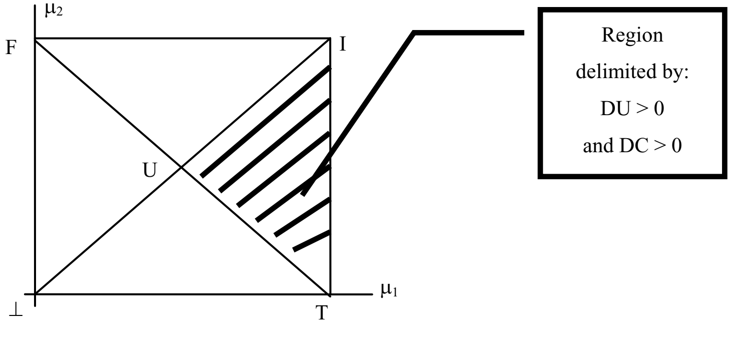


Figure 5 - Unitary Square highlighting the union of regions DU > 0 and DC > 0.

In the following section, other delimited regions and the configuration of the USCP are studied for four other notable points.

5. Delimitations of the Regions in the 2vAPL USCP

Four more lines can be obtained in the USCP according to Figure 6. The new notable points become:

q T - ordered pair (1;0.5) ⇒ almost True point;

q F - ordered pair (0.5;1) ⇒ almost False point;

q¬T - ordered pair (0;0.5) ⇒ almost Non-True point;

q¬F - ordered pair (0.5;0) ⇒ almost Non-False point.

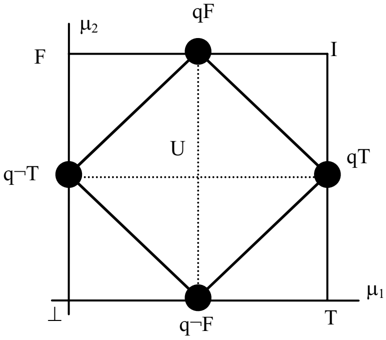


Figure 6 - Lattice represented in the USCP with additional points.

With the addition of these notable points, four new lines appear to delimit the regions in the lattice. To make things easier, each line receives a name according to the proximity to the extreme points in the lattice. The extreme points are: True state, False state, Inconsistent state and Paracomplete state.

Figure 7 shows delimited regions corresponding to each resulting logical state determined by the Degrees of Belief and Disbelief, and it also shows the Totally Inconsistent and Totally Paracomplete regions as well as the regions of Total Falsehood and Total Truth.

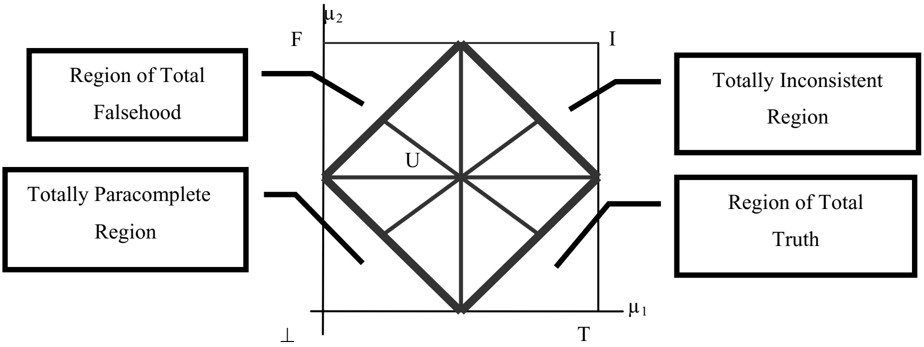

μ_2
F
I
Region of Total Falsehood
Totally Inconsistent Region
U
Totally Paracomplete Region
Region of Total Truth
μ_1
⊥
T

Figure 7- USCP highlighting the Specific Regions.

The delimited regions become:

If DU > 1/2, then the output is Totally Inconsistent.

If |DU| > 1/2, then the output is Totally Paracomplete.

If |DC| > 1/2, then the output is Totally False.
If DC > 1/2, then the output is Totally True.

All the regions that represent the extreme logical states of the lattice: T, F, I and ⊥ are described from this analysis.

6. Representation of the Lattice with the Values of the Degrees of Certainty and of Uncertainty

The DU is defined as being the value that represents, in the lattice, the relationship between the two extreme logical states called Totally Inconsistent and Totally Paracomplete. This degree varies in the real closed interval [-1,1], shown in Figure 8.

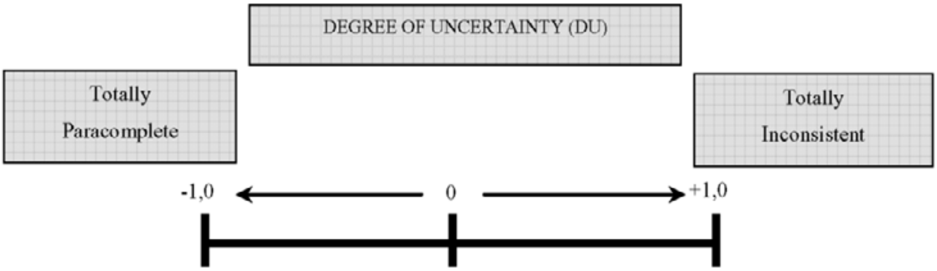


Figure 8 – Representation of the Degree of Uncertainty.

The DC is defined as being the value that represents, in the lattice, the relationship between two extreme logical states called False and True. This degree varies in the real closed interval [-1,1], shown in Figure 9.

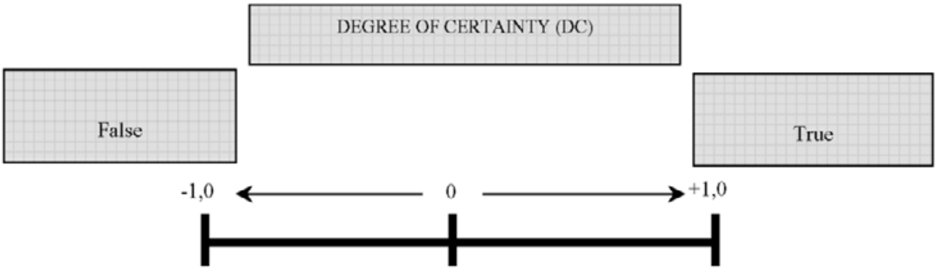


Figure 9 – Representation of the Degree of Certainty.

By overlapping the two axes of the DU with the DC, one may obtain an interrelation of these degrees as shown in Figure 10.

In the graphic representations of the DC and of DU, it is observed that when the DU is greater or equal to ½, the logical state is Totally Inconsistent, according to the delimitations of the regions in Figure 7. Likewise, when the DU presents a value lower or equal to -1/2 the state is Totally Paracomplete.

In the same way, when the DC presents a value greater or equal to 1/2, the logical state is Totally True and when the value is lower or equal to -1/2, the state is Totally False.

All this analysis is done bearing in mind that no other interrelation was considered.

With these considerations, in the graphs of the DC and DU, the values in modules equal or above 1/2 already have well-defined states, and values below 1/2 are interrelated and define regions corresponding to the so-called non-extreme states.

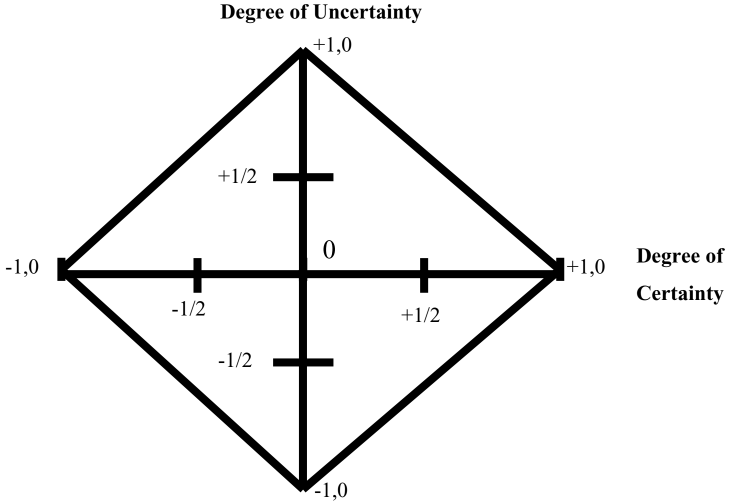


Figure 10 – Representation of interrelated Degrees of Uncertainty and Certainty.

For a better visualization of the description, the certainty and uncertainty control values are named:

C_{csv} - Certainty control superior value;
C_{civ} - Certainty control inferior value;
U_{csv} - Uncertainty control superior value;
U_{civ} - Uncertainty control inferior value.

Thus, for the situation in study:

C_{csv} =	+1/2
C_{civ} =	-1/2
U_{csv} =	+1/2
U_{civ} =	-1/2

Figure 11 shows all the logical states of the lattice with Limit Control Values considered in this configuration.

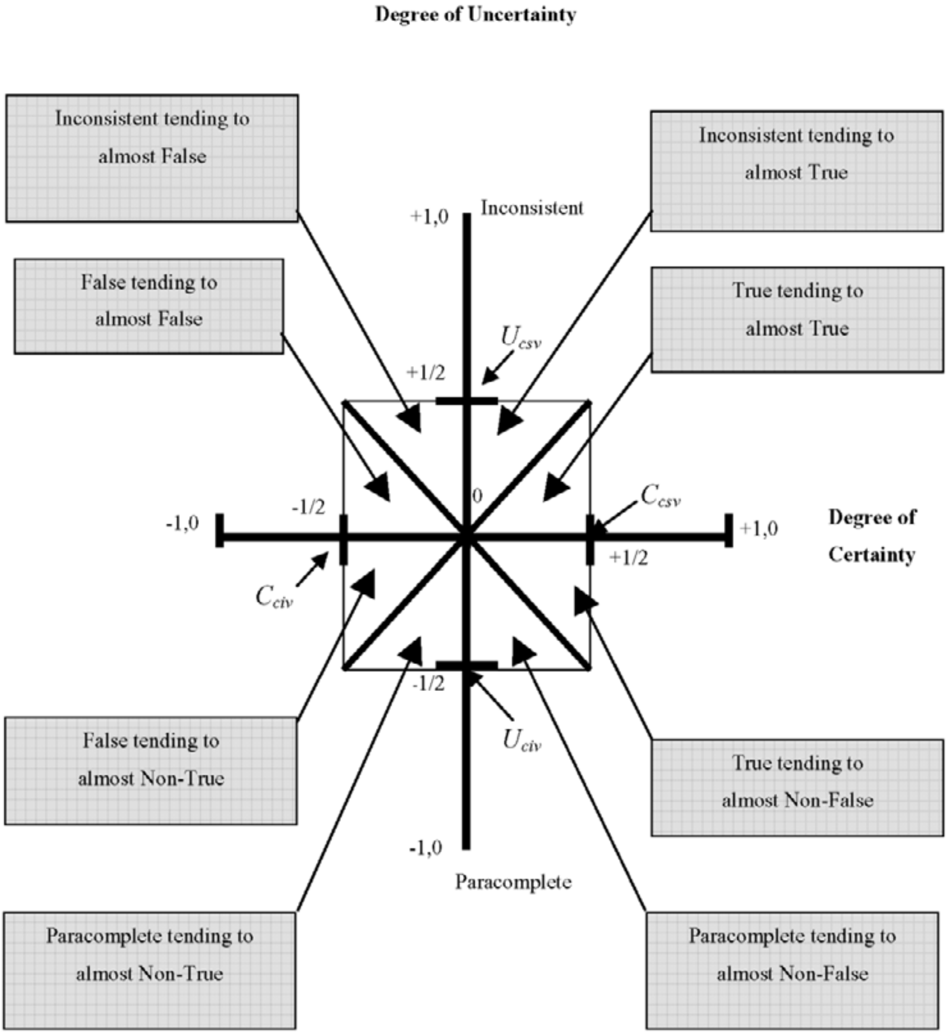

Degree of Uncertainty
Inconsistent tending to almost False
Inconsistent tending to almost True
False tending to almost False
True tending to almost True
+1,0
Inconsistent
U_{csv}
+1/2
0
-1,0
-1/2
C_{csv}
+1,0
+1/2
Degree of Certainty
C_{civ}
-1/2
U_{civ}
False tending to almost Non-True
True tending to almost Non-False
-1,0
Paracomplete
Paracomplete tending to almost Non-True
Paracomplete tending to almost Non-False

Figure 11 – Extreme and non-extreme states with C_{csv}=C_{civ}=1/2 and U_{csv}=U_{civ}=-1/2.

The representations of the extreme and non-extreme states are made through symbols, as shown in Figure 12:

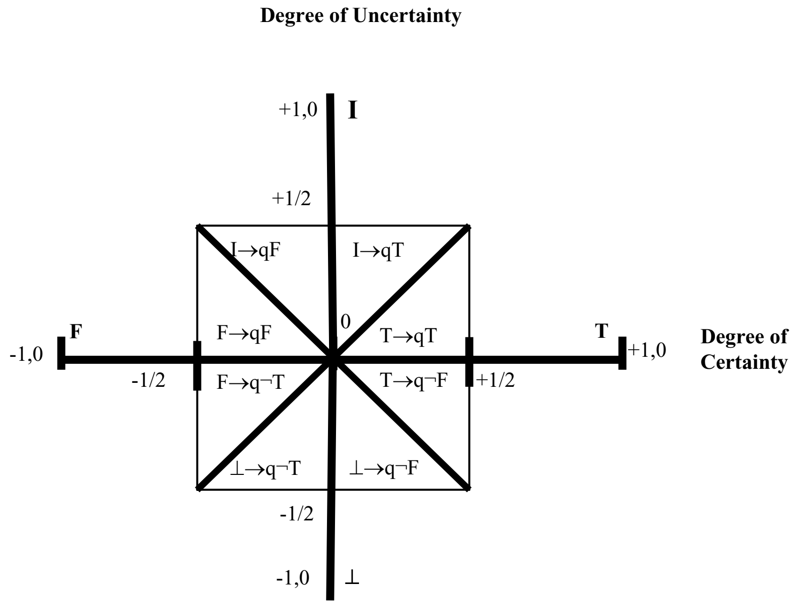


Figure 12 - Symbolical Representation of the extreme and non - extreme states with $C_{csv}=U_{csv}=1/2$ and $C_{civ}=U_{civ}=-1/2$.

The Limit Control Values Ccsv, Cciv, Ucsv and Uciv may be adjusted differently. They are not necessarily equal in module. These values establish the separation among the regions corresponding to the regions of extreme and non-extreme states. Thus, a variation of these values causes a change in the delimited regions in the USCP.

7. Representation of the Degree of Determination

One more degree of interest in the study is added at this point, it is the Degree of Determination (DD), indicated in Figure 13.

The DD is represented by concentric circles at point (0.5; 0.5) - U, which represents the lack of definition point in the USCP. As any point in the plan gets away from the center (U), that is, when the radius of the circumference increases, then the DD increases as well.

Another view is that, any point in the plan gets away from the lack of definition point; the analyzed situation becomes well defined regardless of its result.

Four well defined situations may be obtained. For instance the situation True, False, Paracomplete or Inconsistent. These are situations of maximum DD equal to $\sqrt{2}/2$.

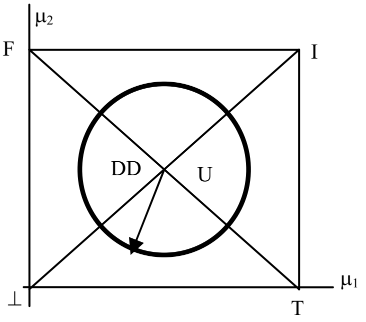


Figure 13 – Representation of the Degree of Determination, DD in the USCP.

The result of the equation that involves the Degrees of Belief, Disbelief, and the Degree of Determination is:

$$DD^2 = (\mu_1 - 0.5)^2 + (\mu_2 - 0.5)^2 \tag{9}$$

For the studied USCP, the DD varies in the real closed interval $[0, \sqrt{2}/2\ ]$.

8. Application of the Negation Operator in the USCP

The negation operator, as seen by its name, gives the sense of negation to the studied proposition. In 2vAPL it is obtained as follows.

Let a proposition P, be composed of the Degrees of Belief and Disbelief $P(\mu_{1P}, \mu_{2P})$. The following proposition $\neg P(\mu_{1P}, \mu_{2P}) = P(1\text{-}\mu_{1P}, 1\text{-}\mu_{2P})$ is obtained by applying the negation operator.

The propositions composed of $\mu_1 = 0.6$ and the $\mu_2 = 0.2$, in a configuration of the USCP is analyzed as an example. The negation operator is then applied and the results of the analysis are seen in the delimited regions that define the resulting output logical states.

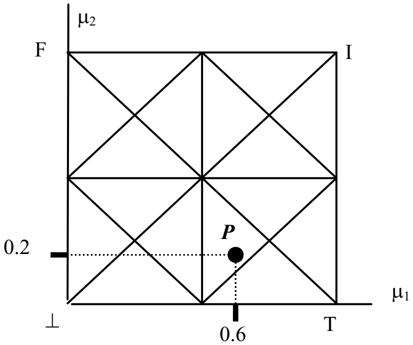


Figure14 – Representation in the USCP of a proposition with the μ_1=0.6 and μ_2=0.2.

In Figure 14, it can be verified that the proposition P(0.6, 0.2) results in a delimited region T→q¬F in the USCP, according to the representation of Figure 12.

It can be observed that any resulting point in this region implies in the resulting output state called True tending to almost Non-False as a final result of the analysis.

By applying the negation operator, the proposition becomes:

$$\neg P\ (0.6, 0.2) = P\ (1\text{-}0.6, 1\text{-}\ 0.2) = P\ (0.4, 0.8)$$

The region of this process is shown in Figure 15.

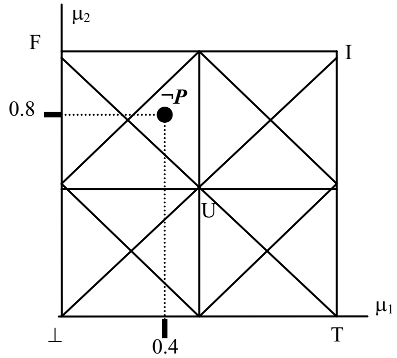


Figure 15 – Application of the negation operator to a proposition P(0.6, 0.2) obtaining P(0.4, 0.8).

It can be observed that the new point has shifted region, located in the resulting output state called False tending to almost False.

8.1. Method to Obtain the Negation Operator from the DU and DC

The method that carries out the logic negation operation by using the DC and DU is described in this section. The changes in the resulting output states when the negation operator is applied can be seen through the lattice of the 2vAPL with the values of the DU and DC.

Using the same values as the previous analysis:

$$\mu_1 = 0.6 \text{ and } \mu_2 = 0.2$$

Then the DU and DC become:

$$DU = \mu_1 + \mu_2 - 1 = 0.6 + 0.2 - 1 = -\ 0.2$$
$$DC = \mu_1 - \mu_2 = 0.6 - 0.2 = 0.4$$

It can be seen in Figure 16 the analysis of resulting states in the lattice with values of DC and DU.

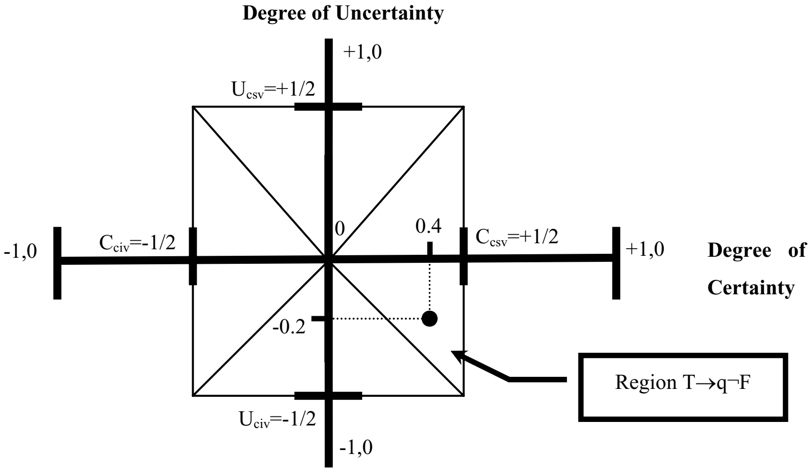


Figure16 – Representation of the DU and DC of the proposition with μ_1=0.6 and μ_2= 0.2.

In Figure 16, the analysis of the values of the Degrees of Belief and Disbelief resulted in a region of the output logical state T→q¬F which is similar to the one obtained though the analysis in the USCP presented before according to Figure 12.

By applying the negation operator, it gives that:

$$\mu_1 = 1 - 0.6 = 0.4 \text{ and } \mu_2 = 1 - 0.2 = 0.8$$

Then, the DU and DC become:

$$DU = \mu_1 + \mu_2 - 1 = 0.4 + 0.8 - 1 = 0.2$$
$$DC = \mu_1 - \mu_2 = 0.4 - 0.8 = - 0.4$$

Figure 17 shows the output logical state considering the new values of the DU and DC.

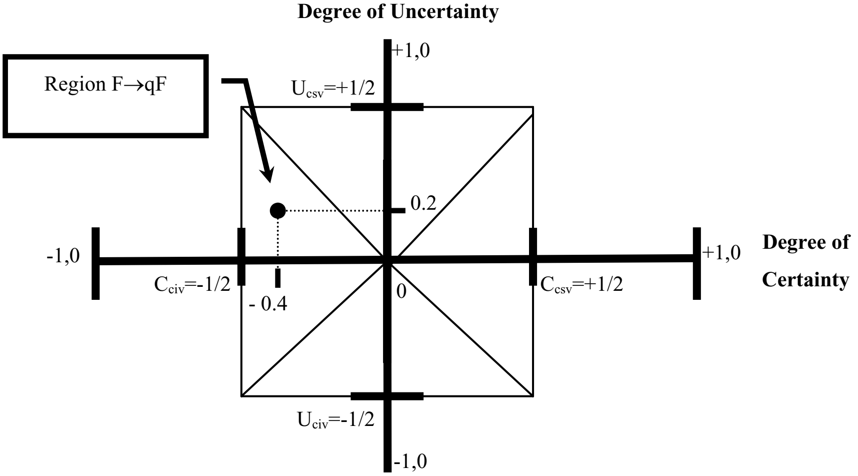


Figure 17 – Representation of the DU and DC of the proposition with μ_1=0.4 and μ_2= 0.8.

It can be observed that by applying the negation operator the result of the new proposition is found in the region corresponding to the output logical state called False tending to almost False (F→qF), whose value is identical to the one found in the previous analysis in the USCP, according to Figure 15.

From the DU and DC encountered, the following observation is clear: the DU and DC presented the same value in module, however with shifted signals.

It can be concluded that:

“To obtain a logic negation in the resulting output logical states in Two-valued Annotated Paraconsistent Logic, the polarity (positive, negative) of the DU and DC should be changed and a paraconsistent analysis on the signals be performed”.

9. Comparative Analysis of 2vAPL Conjunction and Disjunction Functions

According to [2], the connectives of disjunction (OR - $\vee$) and of conjunction (AND-$\wedge$) were duly described:

VI ($P \vee Q$) = 1 if and only if VI (P) = 1 or VI (Q) = 1
VI ($P \wedge Q$) = 1 if and only if VI (P) = 1 and VI (Q) = 1, respectively.

Where P and Q are propositions on which the connectives are applied. These propositions present two annotated signals, each of which is composed of their respective Degrees of Belief and Disbelief.

Consider the first annotated signal as:

Signal P: $P_P(\mu_{1P}, \mu_{2P})$

It follows that: μ_{1P} = Degree of Belief of signal P;
μ_{2P} = Degree of Disbelief of signal P.

And the second annotated signal as:

Signal Q: $P_Q(\mu_{1Q}, \mu_{2Q})$

It follows that: μ_{1Q} = Degree of Belief of Q;
μ_{2Q} = Degree of Disbelief of Q.

The analysis of the application of the connectives $\vee$ (OR) and $\wedge$ (AND) through the USCP of the lattice of the DU and DC follows the procedure:

a) The connectives applied to the input signal of P and Q present:

- In the case of the connective $\vee$ (OR) – Maximization between the Degrees of Believe (μ_{1P}, μ_{1Q}) and Minimization between the Degrees of Disbelief (μ_{2P}, μ_{2Q}).
- In the case of the connective $\wedge$ (AND) - Minimization between the Degrees of Believe (μ_{1P}, μ_{1Q}) and Maximization between the Degrees of Disbelief (μ_{2P}, μ_{2Q}).

b) After Maximization (Minimization) a resulting intermediate signal is obtained: (μ_{1R} , μ_{2R}).

c) An analysis in the USCP or in the lattice with the DU and DC is done with the two resulting signals to obtain the resulting output logical state.

To exemplify the analysis, two annotated logical signals are applied to a system that performs the function of the connective ∨ (OR).

The values of the input signal are:

Signal P: $P_P\ (\mu_{1P}\ ,\ \mu_{2P})$
It follows that: $\mu_{1P} = 0.4$ and $\mu_{2P} = 0.8$

Signal Q: $P_Q\ (\mu_{1Q}\ ,\ \mu_{2Q})$
It follows that: $\mu_{1Q} =0.6$ and $\mu_{2Q} = 0.2$

By applying the formula of the connective ∨ (OR) between propositions P and Q:

$P_P \vee P_Q = (\text{Max}(\mu_{1P}, \mu_{1Q}), \text{Min}(\mu_{2P}, \mu_{2Q})) = (\text{Max}(0.4\ , 0.6),\text{Min}(0.8\ , 0.2))$
$P_P \vee P_Q = (0.6\ , 0.2)$

With: $\mu_{1R} = 0.6$ - Resulting Degree of Belief;
$\mu_{2R} = 0.2$ - Resulting Degree of Disbelief.

With the values of the resulting Degrees of Belief and Disbelief, an analysis in the USCP may be done to obtain the resulting output logical state. Figure 18 shows the situation mentioned above.

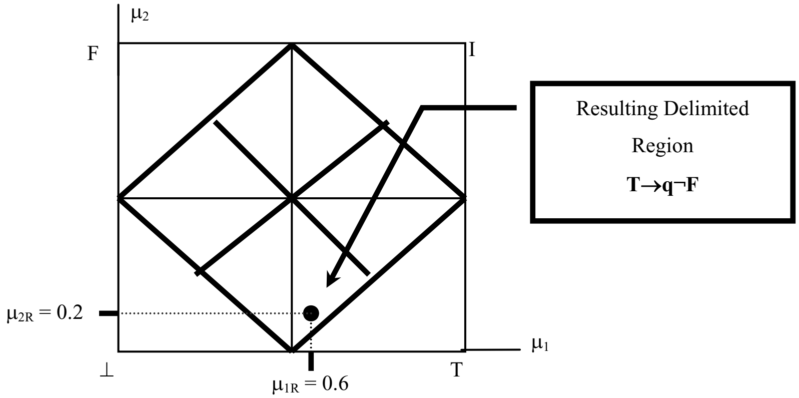


Figure 18 – Representation, in the USCP, of the resulting delimited region, after the action of the connective ∨ (OR) on two input annotated signal.

Now the connective ∧ (AND) is applied to the same propositions P and Q:

Signal P: $P_P\ (\mu_{1P}\ ,\ \mu_{2P})$
It follows that: $\mu_{1P} = 0.4$ and $\mu_{2P} = 0.8$

Signal Q: P_Q (μ_{1Q} , μ_{2Q})
It follows that: μ_{1Q} =0.6 and μ_{2Q} = 0.2

$P_P \wedge P_Q$ = (Min(μ_{1P}, μ_{1Q}), Max(μ_{2P}, μ_{2Q})) = (Min(0.4 , 0.6),Max(0.8 , 0.2))
$P_P \wedge P_Q$ = (0.4 , 0.8)

With: μ_{1R} = 0.4 - Resulting Degree of Belief;
μ_{2R} = 0.8 - Resulting Degree of Disbelief.

With the values of the resulting Degrees of Belief and Disbelief, an analysis on the USCP may be done to obtain the resulting output logical state. Figure 19 shows the mentioned situation.

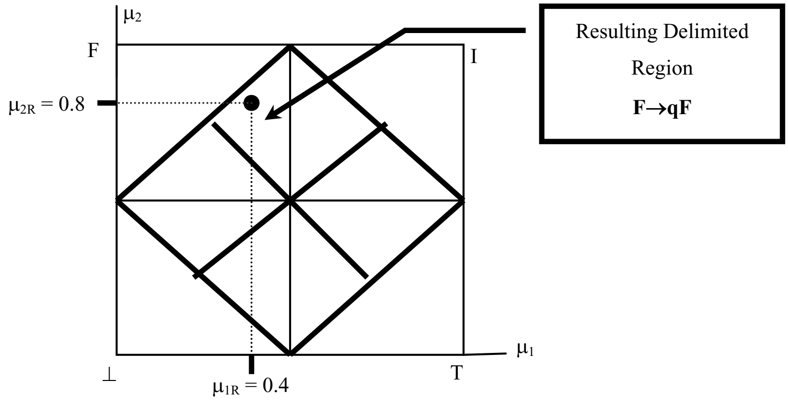


Figure 19 – Representation in the USCP of the resulting delimited region after the action of the connective ∧ (AND) on two annotated input signals.

A practical method to obtain the output states when the connectives ∨ (OR) and ∧ (AND) are applied is done in the USCP.

The annotated signals are composed of the Degrees of Belief and Disbelief, which may be of variable intensity with the time (analogical) or multivalued, according to what was mentioned in the beginning of this chapter. This method of application of the connectives ∨ (OR) and ∧ (AND) is applied to several signal annotated simultaneously. This will be described as follows.

10. Method to Obtain the Output Logical States after the Application of the Connectives ∨ (OR) and ∧ (AND)

Any proposition P is composed of a Degree of Belief and Disbelief when analyzed in the USCP, such that it results in a point located in a delimited region and relates to a single resulting output logical state. Thus, for any two propositions P and Q, there results in two points according to Figure 20.

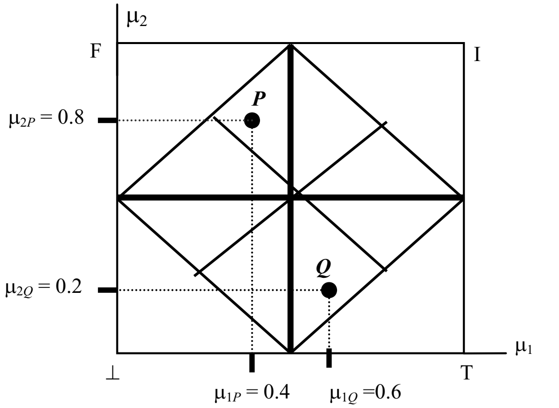


Figure 20 – Representation in the USCP of two propositions P(0.4, 0.8)e Q(0.6, 0.2).

10.1. Application of the Connective ∨ (OR):

The connective ∨ (OR) performs the Maximization between the values of the Degrees of Belief and performs the Minimization between the values of the Disbelief between two propositions. To find the point resulting from the application of the connective ∨ (OR) one must construct a rectangle with the sides parallel to the cartesian coordinates. The two propositions are the diagonally opposite vertices.

The result of the disjunction will be the right inferior vertex of the rectangle. This represents the fact that the successive application of the operator ∨ (OR) tends to draw the result closer to the notable point notable Truth.

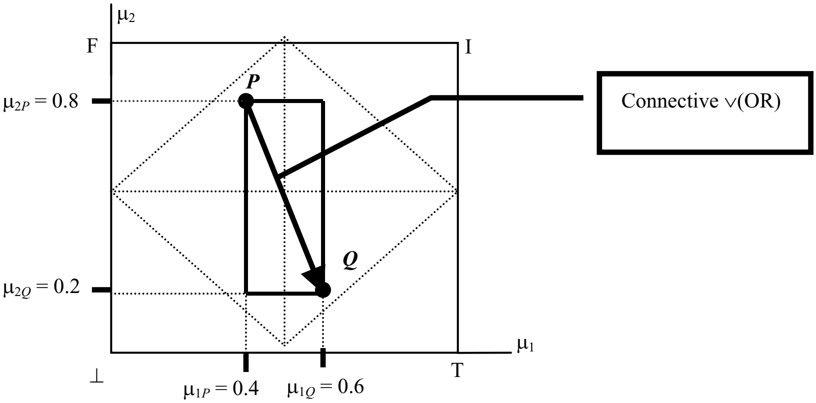


Figure 21 – Representation in the USCP of the method of application of the connective ∨(OR) between two propositions P(0.4, 0.8) and Q(0.6, 0.2).

The application of the connective ∨ (OR) on the signals P and Q resulted in a point corresponding to the resulting output logical state True tending to almost Non-False according to Figure 21 above, which corresponds to Figure 18. In the USCP, the visualization of the application of the connective becomes simpler, making the application easier to several signals at the same time. One must find the Degree of Belief of greater signal and the Degree of Disbelief of lower signal and prolong these values to meet the intersection. This is the logical state resulting point.

In this way, the connective ∨ (OR) when applied to two annotated signals, one True the other False, the resulting logical state is True. Therefore, the prevailing logical state in the application of the connective ∨ (OR) is the state True.

10.2. Application of the Connective ∧ (AND):

The connective ∧ (AND) performs the Minimization between the values of the Degrees of Belief and the Maximization between the values of the Degrees of Disbelief between two propositions. To find the point resulting from the application of the connective ∧ (AND), one must construct a rectangle with the sides parallel to the cartesian coordinates. The two propositions are the diagonally opposite vertices.

The result of the conjunction will be the left superior vertex of the rectangle. This represents the fact that the successive application of the operator ∧ (AND) tends to draw the result closer to the notable point False.

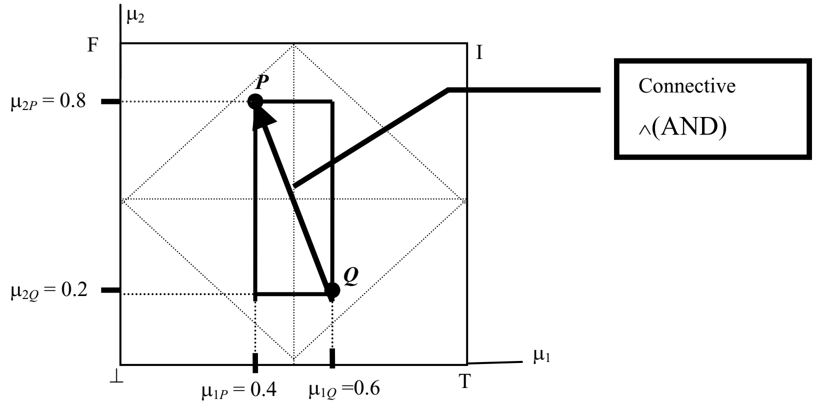


Figure 22 – Representation in the USCP of the method of application of the connective ∧ (AND) between propositions P(0.4, 0.8) and Q(0.6, 0.2).

The application of the connective ∧ (AND) to signals P and Q resulted in a point that corresponds to the resulting output logical state of False tending to almost False according to Figure 22, which corresponds to Figure 19.

In the USCP, the visualization of the application of the connective becomes simpler, making the application easier to several signals at the same time. One must find the Degree of Belief of lower value signal and the Degree of Disbelief of greater value signal, and prolong these values to meet the intersection. This is the logical state resulting point.

In this way, the connective ∧ (AND) when applied to two annotated signals, one True the other False, the resulting logical state is False. Therefore, the prevailing logical state in the application of the connective ∧ (AND) is the False state.

With these methods mentioned above, the analysis of 2vAPL in the USCP becomes easier to see and shows that the application of the Negation operator, the Disjunction Connective (OR), and the Conjunction Connective (AND) may be applied directly without the need of Truth Tables.

11. Obtaining the Resulting Signals

Figures 23, 24, and 25 illustrate the procedure to obtain the disjunction and conjunction connectives as well as the negation operator.

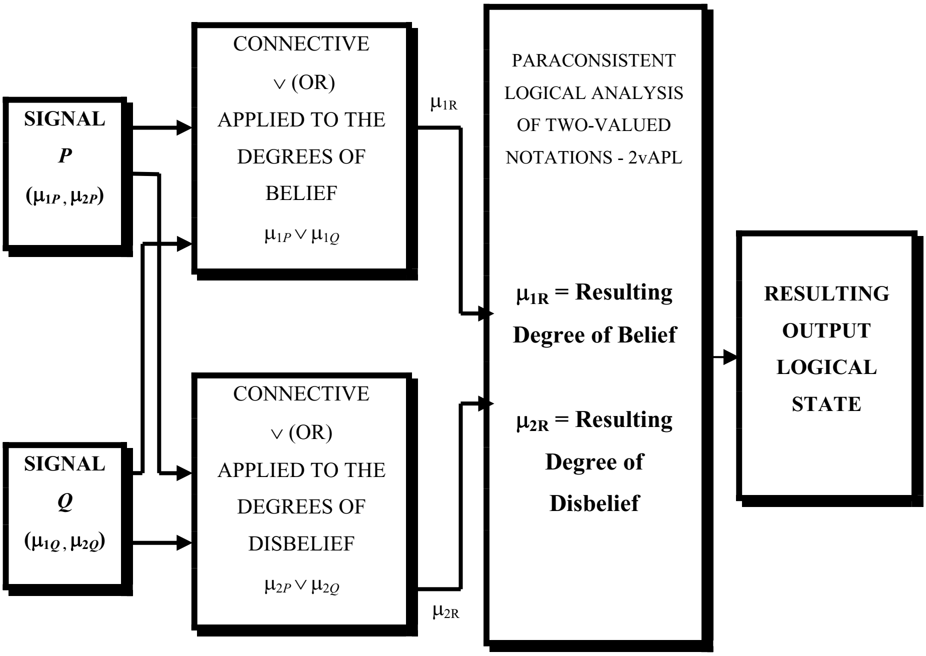


Figure 23 – Representation of the procedures to obtain the function of the connective ∨ (OR) in the 2vAPL.

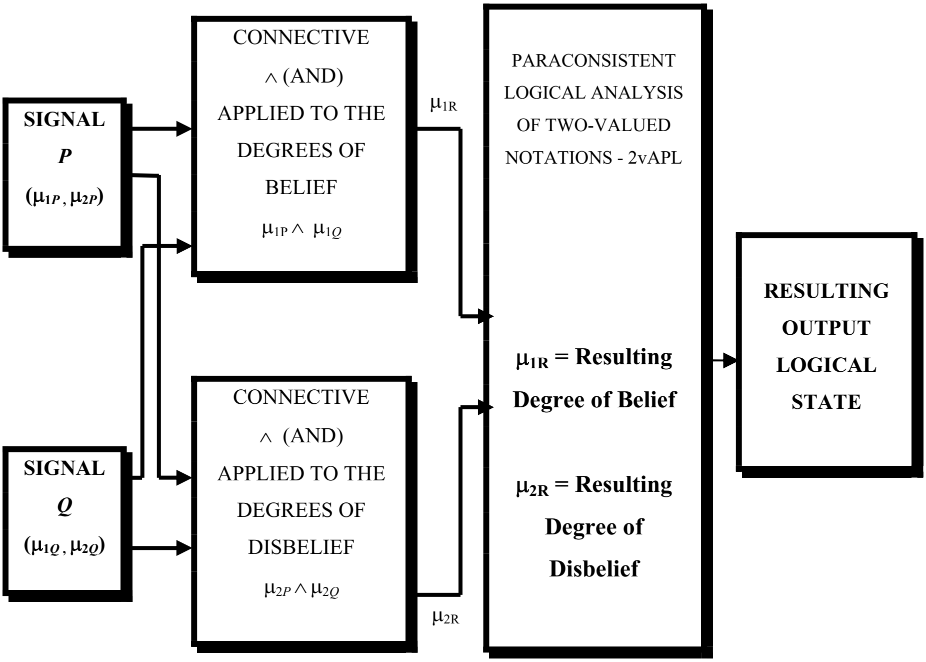


Figure 24 – Representation of the procedures to obtain the function of the connective $\wedge$ (AND) in 2vAPL.

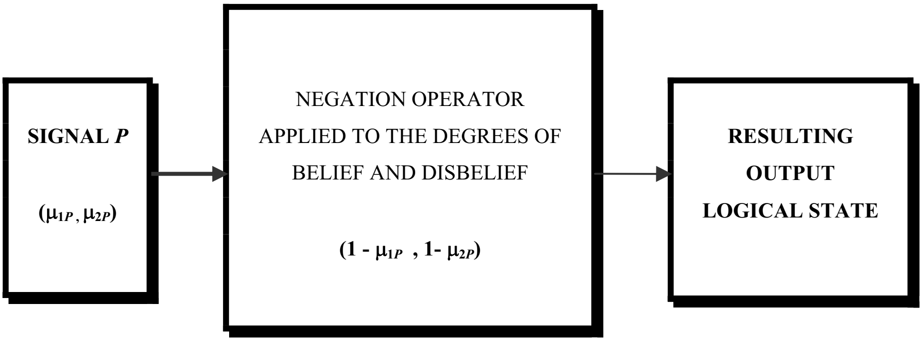


Figure 25 - Representation of the procedures to obtain the function of the negation operator in 2vAPL.

The application of the connectives and the negation operator becomes simpler by using the procedures exposed above.

12. Three-valued Annotated Paraconsistent Logic - 3vAPL

A perpendicular axis is introduced in the USCP. It is interpreted as the Degree of Expertise (e), according to [11] and [15], such that 2vAPL becomes extended to 3vAPL.

The values of e vary in the real closed interval [0,1], like the Degrees of Belief and Disbelief. Thus, a point obtained from the triple (μ_1,μ_2,e) is interpreted in the Unitary Analyzer Cube, shown in Figure 26.

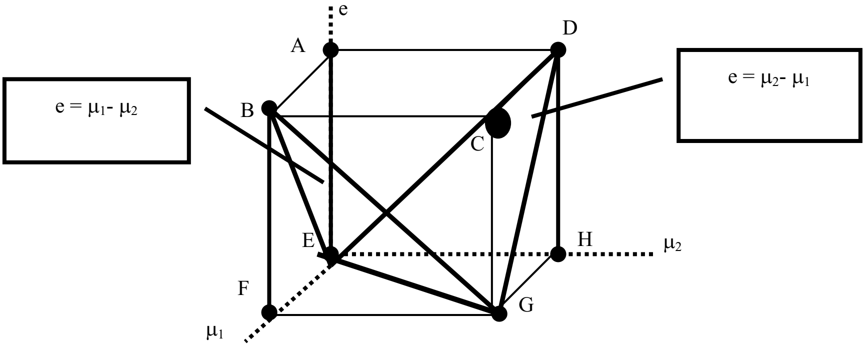


Figure 26 - Unitary Analyzer Cube.

For plan e=1, the Degree of Expertise is maximum. This is referred to as Expert. For plan e=0, the Degree of Expertise is minimum. This is referred to as Neophyte. For the intermediary plans, the Degrees of Expertise vary in the real closed interval [0,1].

New notable points on the Unitary Cube are determined, according to Figure 26.

Point A = (0,0,1)⇒ Expert with paracomplete information, ⊥;

Point B = (1,0,1)⇒ Expert opting for diagnostics referring to axis x, D_x;

Point C = (1,1,1)⇒ Expert with inconsistent information, I;

Point D = (0,1,1)⇒ Expert opting for diagnostics referring to axis y, D_y;

Point E = (0,0,0)⇒ Neophyte with paracomplete information, ⊥;

Point F = (1,0,0)⇒ Neophyte opting for diagnostics referring to axis x, D_x;

Point G = (1,1,0)⇒ Neophyte with inconsistent information, I;

Point H = (0,1,0)⇒ Neophyte opting for diagnostics referring to axis y, D_y;

Plan e = μ_1- μ_2⇒ Limit Plan of the classical case, D_x;

The points below this plan and limited by the cube are points that determine the diagnostics referring to axis x. Points above are in a "Biased Region";

Plan $e = \mu_2 - \mu_1 \Rightarrow$ Limit Plan of the classical case, D_y;

The points below this plan and limited by the cube are those that determine the diagnostics referring to axis y. Points above are in a "Biased Region";

However, restricted to the ordered pair (μ_1,μ_2), one observes that the result of the studies carried out in references [13], [14] and [15], is a particular case of *Degree of Expertise* e = 0.5, seen in Figure 26. Thus, all the analyzed considerations may be expanded according to the increase or decrease of the Degree of Expertise.

12.1. Delimitations of Regions in the Unitary Analyzer Cube

According to Figure 27, the Unitary Cube is delimited into Diagnostic Regions.

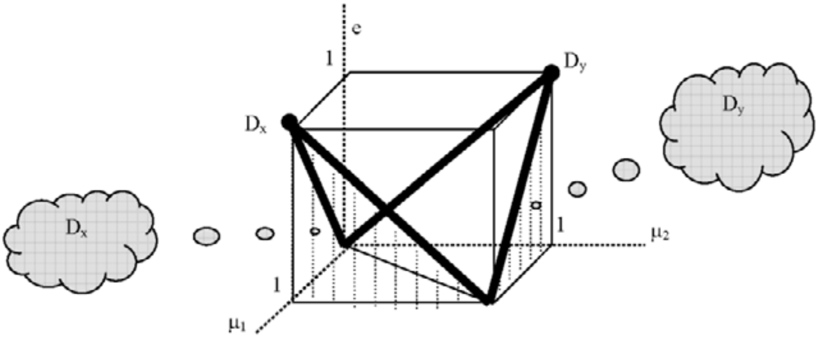


Figure 27 – Representation of the Diagnostic Regions.

Point (1,0,1), on plan e = 1, may be interpreted as an expert deciding for a diagnostics referring to axis x, D_x. While, on plan e = 0, the points limited by the diagonal $\mu_1 = \mu_2$ and by axis x, may be interpreted as being a neophyte deciding for the diagnostic D_x. This is the larger region referring to D_x due to, exactly, the neophyte's lack of experience.

As the Degree of Expertise increases, the region for D_x becomes restricted until it gets to point (1,0,1) analyzed above.

In Figure 28, the Unitary Cube is delimited into Inconsistent and Paracomplete regions.

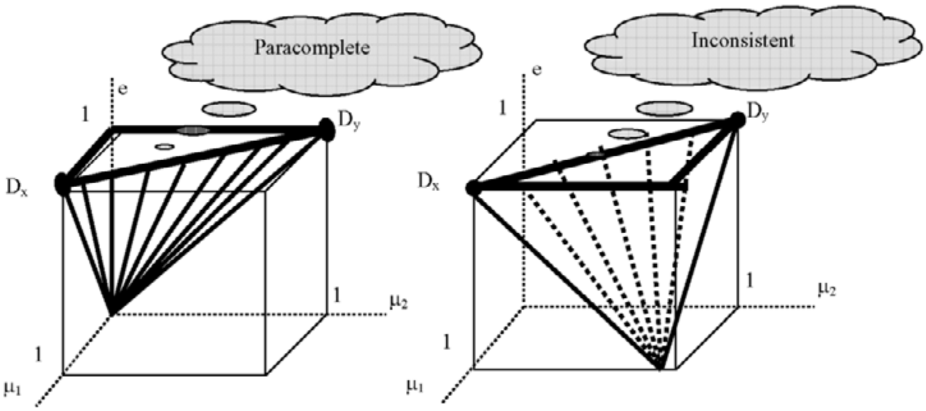


Figure 28 – Representation of the Paracomplete and Inconsistent Regions.

The above regions represent Inconsistent and Paracomplete Regions, regions containing points which permit Inconsistent and Paracomplete interpretations. Points close to (0,0,1) and to (1,1,1) may be interpreted as an Inconsistent and Paracomplete expert, respectively. This situation happens on plan e = 1 where the experts are allowed to opt only for one of the points, (1,0,1) or (0,1,1), which are respectively diagnostic x, D_x or diagnostic y, D_y.

Note that, as the neophyte, at first, is admitted to have contradictory opinions, points (0,0,0) and (1,1,0) are out of the inconsistent and paracomplete regions.

12.2. Interpretations in the Unitary Analyzer Cube

In this section, three different situations are analyzed in the Unitary Analyzer Cube, that is, opinions from three different experts.

The first situation to be analyzed is the situation at the top of the Unitary Cube going toward its base.

For e = 1 there is the following analysis of the regions, according to Figure 29.

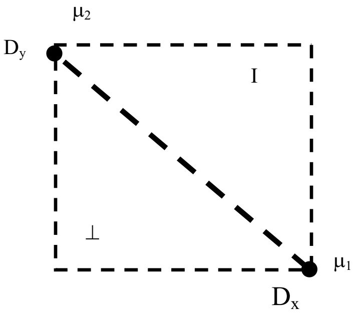


Figure 29 – Analysis for e = 1.

From an expert of degree e = 1, coherent decisions are expected without indecisions or lack of knowledge, or even inconsistencies of any kind. In short, a *great expert* decides only between two diagnostics D_x or D_y, or points (1,0,1) or (0,1,1) respectively. Therefore, the Inconsistent and Paracomplete regions present maximum area.

An expert of degree e = 0.5 is a particular case. The exact situation studied in the previous section. Therefore, the delimited regions are the same and have already been shown. It may be seen from Figure 30 that the Diagnostic D_x and D_y, Inconsistent, and Paracomplete Regions are equal. Therefore, the biased regions present maximum area. The Notable Points tend to displace to the positions of the Inconsistent and Paracomplete Point.

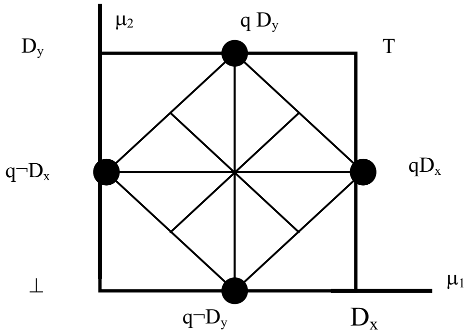


Figure 30 – Analysis for e = 0.5.

The analysis of the last situation shows that for a neophyte whose expertise is e = 0 everything is permitted; there are no restrictions. Considering the inexperience, positions of Inconsistency, Paracompleteness and/or Indetermination are admitted. It can be noticed from Figure 31 that the Diagnostic Regions get to maximum, and the Inconsistent and Paracomplete regions become well defined points, (1,1,0) and (0,0,0) respectively. The Notable Points are also taken to the same points.

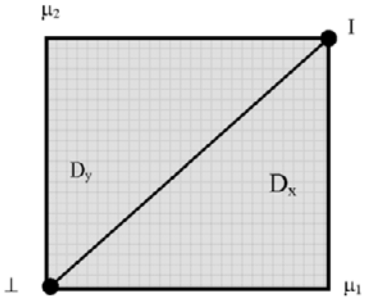


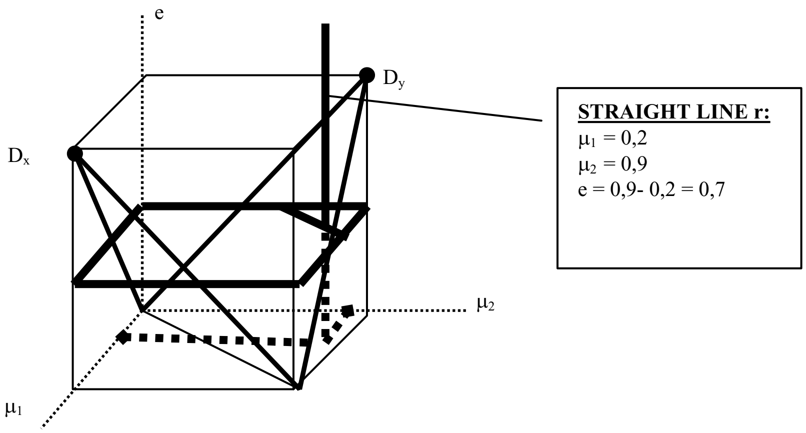
Figure 31 - Analysis for e = 0.

12.3. Method to Obtain the Degree of Expertise from the Degrees of Belief and Disbelief

As it was seen before, the plans π_1: e = μ_1- μ_2 and π_2: e = μ_2- μ_1, relate mathematically the Degrees of Belief, Disbelief, and Expertise.

Consider a proposition P, composed of the Degrees of Belief, $\mu1$ and Disbelief, $\mu2$, such that it results in a point located on plan $\mu1\mu2$. To find the value of the Degree of Expertise for proposition P, draw a line parallel to Expertise axis, from the values of $\mu1$ and $\mu2$. The intersection of line r with one of the plans, $\pi1$ or $\pi2$, establishes any plan e, parallel to plan e = 0, determining thus, the value of the Degree of Expertise of 3vAPL.

Figure 32 shows the method to obtain the Degree of Expertise in 3vAPL. The *Para-Expert* algorithm, which defines all the regions of the Unitary Analyzer Cube, is found in reference [9].



STRAIGHT LINE r:
μ_1 = 0,2
μ_2 = 0,9
e = 0,9- 0,2 = 0,7

Figure 32 – Obtaining the Degree of Expertise in 3vAPL.

13. Four-valued Annotated Paraconsistent Logic – 4vAPL

A point may be analyzed by moving along the Unitary Analyzer Cube, as shown in Figure 33, according to [8] and [12].

At time t1, the point is found on position s1, at time t2, the point is found on position s2, in such a way that as time flows the point describes a curve C in the interior of the Unitary Analyzer Cube.

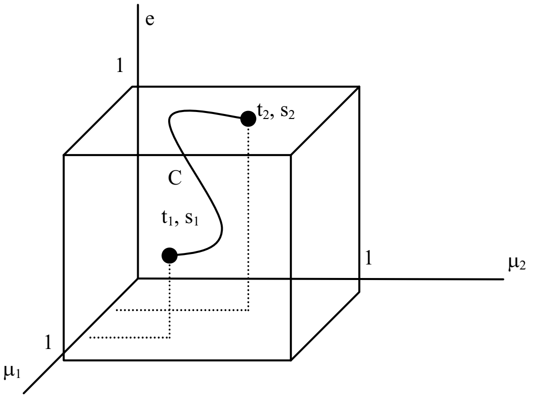


Figure 33 – Temporality in the 3vAPL Unitary Cube.

This behavior allows the introduction of one more variable, time t, to Three-valued Annotated Paraconsistent Logic- 3vAPL; thus extending it to Four-Valued Annotated Paraconsistent Logic- 4vAPL. In 4vAPL, the point in the Unitary Cube is represented by a quadruple (μ1, μ2, e, t). The intention of introducing one more annotated variable to represent the point is to be able to analyze the behavioral evolution of the Experts.

Thus, a neophyte (expert of degree e=0) regarding their inexperience, will adquire experience as the variable time flows. Their Degree of Expertise is expected to increase so as to be able to decide between the two diagnostic D_x or D_y. Roughly speaking, until it is found at the top of the Unitary Cube with the behavior of a ‘classical case’. This analysis may be done for any level of Expertise. The essence of the fourth dimension time is to evaluate the behavior of the experts in decision-making of a certain system.

14. Conclusion

This work has presented an extension of Two-valued Annotated Paraconsistent Logic – 2vAPL, in such a way that if the propositions are well-behaved, then all the valid formula classical calculus must remain valid. This allows the modelling of systems to keep their classical existing features. It should, at the same time, enable the application of Paraconsistent Logic.

Concerning the extension of 2vAPL, new lines in the USCP were introduced as weel as new notable points, influencing the determination of the regions in the lattice. Also, new application methods for the disjunction and conjunction connectives as well as a new proposal of application of the negation operator were addressed so as to approach classical logic.

Two-valued Annotated Paraconsistent Logic - 2vAPL was extended to Three-valued Annotated Paraconsistent Logic – 3vAPL with the introduction of the Unitary Analyzer Cube. The Unitary Analyzer Cube was built from the values of the Degrees of Belief and Disbelief in 2vAPL, determining thus, the third degree, Degree of Expertise. This includes the opinions of experts to describe the problems more closely to real conditions. The inclusion of the Degree of Expertise provides support to decision-making process between two diagnostic, D_x and D_y.

Consequently, an extention of Three-valued Annotated Paraconsistent Logic - 3vAPL to Four-valued Annotated Paraconsistent Logic – 4vAPL was performed with the introduction of a fourth notation time, t. This new notation enables the assessment of the experts' behavior within time. It describes the situations closer to reality.

Acknowledgments

The authors would like to thank CNPq, CAPES, and FAPEMIG - Brazilian research funding agencies, for the research scholarships, which supported this work.

References

[1] Pearl, J., *"Artificial Intelligence"*, Belief networks revisited, vol. 59, pp. 49-56, Amsterdam, 1993.

[2] Abe, J. M., *Foundations of Annotated Logic*, (in Portuguese) Ph. D. Thesis, FFLCH/USP, SP, 1992.

[3] Da Costa, N. C. A. & Hensche, L. J. & Subrahmanin, V. S., *Automatic Theorem Proving in Paraconsistent Logics: Theory and Implementation*, Coleção Documentos n° 3, USP, SP, 1990.

[4] Subrahmanian, V. S., *On the Semantics of Quantitative Logic Programs,* Proc. 4th IEEE Symposium on Logic Programming, Computer Society Press, Washington DC, 1987.

[5] Da Costa, N. C. A .& Abe, J. M. & Subrahmanian, V. S., *"Remarks on Annotated Logic",* Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik, Vol. 37, pp.561-570, 1991.

[6] Da Costa, N. C. A. & Subrahmanian, V. S. & Vago, C. *"The Paraconsistent Logic $P\tau$"*, Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik, Vol. 37, 139 - 148, 1991.

[7] Da Costa, Newton C. A., The Philosophical *Importance of Paraconsistent Logic*, Bol. Soc. Paranaense de Matemática, vol. 11, nº 2, 1990.

[8] Da Costa, Newton C. A., *On the Theory of Inconsistent Formal Systems*, Notre Dame Journal of Formal Logic, vol, 11, 1974.

[9] Da Costa, Newton C. A., Abe, J. M., Murolo, A. C., "Applied Paraconsistent Logic", Atlas Publishing, 1999.

[10] Da Silva Filho, J. I., "Methods of Applications of Two-valued Annotated Paraconsistent Logic - 2vAPL with Algorithm and Implementation of Electronic Circuits", (in Portuguese) Ph. D. Thesis, EPUSP, SP, 1999.

[11] Martins, H. G., Lambert-Torres, G., Pontin, L. F., "An Annotated Paraconsistent Logic", COMMUNICAR Publishing, 2007 (in Portuguese).

[12] Kleene, Stephen C., *"Introduction to Metamathematics"*, Wolters-Noordhoff Publishing - Groningen, North-Holland Publishing Company - Amsterdam, pp 332-340, 1952.

[13] Lambert-Torres, G., Costa, C. I. A., Gonzaga, Helga, "Decision- Making System based on Fuzzy and Paraconsistent Logics", Logic, Artificial Intelligence and Robotics – Frontiers in Artificial Intelligence and Applications, LAPTEC 2001, Ed. Jair M. Abe and João I. da Silva Filho – IOS Press, pp. 135-146, 2001.

[14] Martins, H. G., Lambert-Torres, G., Pontin, L. F., "Extension from NPL2v to NPL3v", Advances in Intelligent Systems and Robotics, by G. Lambert-Torres, J.M. Abe, M.L. Mucheroni & P.E. Cruvinel, IOS Press, ISBN 1 58603 386-7, Vol. I, pp. 9-17, 2003.

[15] Martins, H. G., "A Four-valued Annotated Paraconsistent Logic – 4vAPL Applied to CBR for the restoration of Electrical Substation", (in Portuguese) Ph. D. Thesis, UNIFEI, Brazil, 2003.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-114

Creation of Virtual Environments through Knowledge-Aid Declarative Modeling

Jaime ZARAGOZA [a], Félix RAMOS [a], Héctor Rafael OROZCO [a] and Véronique GAILDRAT [b]

[a] *Centro de Investigación y de Estudios Avanzados del I.P.N. Unidad Guadalajara*
Av. Científica 1145, Col. El Bajío, 45010 Zapopan, Jal., México
[b] *Universitè Paul Sabatier IRIT-CNRS UMR,*
Toulouse Cedex 9, France

Abstract. This article deals with the problem of Declarative Modeling of scenarios in 3D. The innovation presented in this work consists in the use of a knowledge base to aid the parsing of the declarative language. The example described in this paper shows that this strategy reduces the complexity of semantic analysis, which is rather time- and resource-consuming. The results of our work confirm that this contribution is useful mainly when the proposed language constructions have high complexity.

Keywords. Declarative Modeling, Virtual Environment, Declarative 3D editor, Animated Virtual Creatures, Virtual Agents.

Introduction

The use of computer graphics and technologies in fields such as computer games, film-making, training or even education has increased in the last years. The creation of virtual environments that represent real situations, fantasy worlds or lost places can be achieved with the help of several tools, from basic 3D modelers, like Flux Studio [1] or SketchUP [2], to complete suites for creating 3D worlds, such as 3D Max Studio [3] or Maya [4]. Creating complex and detailed worlds like those of films and video games with these tools requires experience [5]. However, frequently those who design scenarios are not experts; they don't have the experience of those who model the scenarios in 3D.

Our aim is to provide final users with a descriptive language which allows them to set a scenario and a scene and leaves the work of their creation for the computer. This approach makes the VR available for final users of different areas, for those who create scenarios of plays, games or simulations of film scenes, for instance.

If the user could just describe the scenarios and/or scenes intended to be created and let the computer generate the environment, that is, the virtual scenario and the elements necessary for animating the scene, the creation of virtual environments could be at the reach of any user. So, even the non-expert users could find in the VR a valuable tool for completing their tasks. The objective of the GeDA-3D project is to facilitate the creation, development and management of virtual environments by both expert and non-expert users. The main objective of this article is to expose the method for generating virtual scenarios by means of the declarative modeling technique, supported by the knowledge base specialized in semantic analysis and geometric conflict solving. We adopt the definition from [6] for:

Definition 1: Declarative modeling is "a technique that allows the description of a scenario to be designed in an intuitive manner, by only giving some expected properties of the scenario and letting the software modeler find solutions, if any, which satisfy the restrictions". This process usually is divided in three phases [7]: *Description* (defines the interaction language); *Scene generation* (the result of one or more scenes the modeler generates, that match with the description introduced by the user); *Insight* (corresponds to results presented to the user; if more than one result is found, the user must choose the solution).

The bases for the creation of the virtual scenario are centered in the declarative modeling method, which is supported by constraint satisfaction problem (CSP) solving techniques. These techniques solve any conflict between the geometry of the model's entities. This procedure is also supported by the knowledge base, which contains all the necessary information to both satisfy the user's request, and validate the solutions obtained by the method.

The following parts of this chapter contain the description of the related works, the description of GeDA-3D architecture (the GeDA-3D is a complete project that includes the present work as one of its parts), our proposal to declarative modeling with the use of ontology, our conclusions and future work.

1. Related Works

The creation of virtual worlds and their representation in 3D can be a difficult task, especially for the users with lack of experience in the field of 3D modeling. The reason is that the tools necessary for modeling are often difficult to manage, and even experienced users require some time to create stunning results, such as those seen in video games or movies.

There are several works focused on declarative modeling. Some of them are oriented to architectonic aspects, others to virtual scenarios generation, sketching or design. Among the tools focused on architectonic development we can highlight the following ones:

The FL-System by Jean-Eudes Marvie et al. [8] specializes in the generation of complex city models. The system's input is defined in a specialized grammar and can generate city models of variable complexity. It is completely based on a System-L variant and uses VRML97 for the visualization of the models.

CityEngine [9], by Yoav I. H. Parish and Pascal Müeller, is a project focused on the modeling of full cities, with an entry composed by statistical data and geographic information. It also bases its design method on the System-L, using a specialized grammar.

However, our interest goes to the works focused on the creation of virtual scenarios, such as:

1.1.WordsEye: Automatic text-to-scene conversion system

Bob Coyne and Richard Asproad have presented WordsEye [10] developed at the AT&T laboratories, a system which allows any user to generate a 3D scene on the basis of description written in human language, for instance: “The bird is in the bird cage. The bird cage is on the chair”. The text is initially marked and analyzed using a part-of-speech tagger and a statistical analyzer. The output of this process is the analysis tree that represents the structure of the sentence. Next, a *depictor* (low level graphic representation) is assigned to each semantic element. These depictors are modified to match with the poses and actions described in the text by means of the inverse kinematics. After that, the depictors’ implicit and conflicted constraints are solved. Each depictor then is applied keeping its constraints, to incrementally build the scene, the background environment, the terrain plane. The lights are added and the camera is positioned to finally represent the scene. In the case the text includes some abstraction or description that does not contain physical properties or relations, other techniques can be used, such as: textualization, emblematization, characterization, lateralization or personification. This system accomplishes the text-to-scene conversion by using statistical methods and constraints solvers, and also has a variety of techniques to represent certain expressions. However, the scenes are presented in static form, and the user has no interaction with the representation.

1.2.DEM^{2}ONS: High Level Declarative Modeler for 3D Graphic Applications

DEM^{2}ONS has been designed by Ghassan Kwaiter, Véronique Gaildrat and René Caubet to offer the user the possibility to construct easily the 3D scenes in a natural way and with high level of abstraction. It is composed of two parts: *modal interface* and *3D scene modeler* [11]. The modal interface of DEM^{2}ONS allows the user to communicate with the system, using simultaneously several combined methods provided by different input modules (data globes, speech recognition system, spaceball and mouse). The syntactic analysis and dedicated interface modules evaluate and control the low-level events to transform them into normalized events. For the 3D scene modeler ORANOS is used. It is a constraint solver designed with several characteristics that allows the expansion of declarative modeling applications, such as: generality, breakup prevention and dynamic constraint solving. Finally, the objects are modeled and rendered by the Inventor Tool Set, which provides the GUI (Graphic User Interface), as well as the menus and tabs. This system solves any constraint problem, but only allows the creation of static objects, with no avatar support and no interaction between the user and the environment, not to mention modifications in the scenario's description.

1.3.Multiformes: Declarative Modeler as 3D sketch tool

William Ruchaud and Dimitri Plemenos have presented Multiformes [12], a general purpose declarative modeler, specially designed for sketching 3D scenarios. As it is by any other declarative modeler, the input of MultiFormes contains description of the scenario to create (geometric elements’ characteristics and the relationships between them). The most important feature of MultiFormes is its ability to explore automatically all the possible variations of the scenario description. This means that

Multiformes presents several variations of the same scenario. This can lead the user to choose a variation not considered initially. In Multiformes, a scenario's description includes two sets: the set of geometric objects present in the scenario, and the set of relationships existent between these geometric objects. To allow the progressive refinement of the scenario, MultiFormes uses hierarchical approximations for the scenario modeling. This approach allows describing the scenario incrementally to obtain more or less detailed descriptions. The geometric restriction solver is the core of the system and it allows exploring different variations of a sketch. Even when this system obtains its solutions in incremental ways and is capable of solving the constraints requested, the user must tell the system how to construct the scenario using complex mathematical sets, as opposed to our declarative creation, where the user formulates simple states in a natural-like language only describing what should be created.

1.4.CAPS: Constraint-Based Automatic Placement System

CAPS is a positioning system based on restrictions [13], developed by Ken Xu, Kame Stewart and Eugene Fiume. It models big and complex scenarios using a set of intuitive positioning restrictions, which allows to manage several objects simultaneously. Pseudo-physic is used to assure stable positioning. The system also employs semantic techniques to position objects, using concepts such as fragility, usability or interaction between the objects. Finally, it allows using input methods with high levels of freedom, such as Space Ball or Data Glove. The objects can only be positioned one by one. The direct interaction with the objects is possible while maintaining the relationships between them by means of pseudo-physics or grouping. The CAPS system is oriented towards scenario visualization, with no possibilities for self-evolution.

None of the previously described systems allows evolution of the entities or environments created by the user. All these works rely on specialized data input methods, either specialized hardware or input language, with the sole exception of WordEyes. Its output is a static view of the scenario created by means of natural language, but in spite of it, its representation abilities are limited, that is to say, it is not possible just to describe goals. Representing unknown concepts can lead specialized techniques to bizarre interpretations.

1.5.Knowledge Base Creation Tools

There are several ontology creation tools, which vary in standards and languages. Some of them are:

1.5.1.Ontolingua [14].

Developed at the KSL of the Stanford University, consists of the *server* and the *representation language*. The server provides ontology with repository, allowing its creation and modification. Ontologies in the repository can be joined or included in a new ontology. Interaction with the server is conducted by the use of any standard web browser. The server is designed to allow cooperation in ontology creation, that is to say, easy creation of new ontologies by including (parts of) existing ontologies from the repository and the primitives from the ontology frame. Ontologies stored on the server can be converted into different formats and it is possible to transfer definitions from the ontologies in different languages into the Ontolingua language. The Ontolingua server can be accessed by other programs that know how to use the ontology store in the representation language [15].

1.5.2.WebOnto [16].

Completely accessible from the internet, it was developed by the Knowledge Media Institute of the Open University and Design to support creation, navigation and collaborative edition of ontologies. WebOnto was designed to provide a graphic interface; it allows direct manipulation and complements the ontology discussion tool Tadzebao. This tool uses the language OCLM (Operational Conceptual Modeling Language), originally developed for the VITAL project [17], to model ontologies. The tool offers a number of useful characteristics, such as saving structure diagrams, relationships view, classes, rules, etc. Other characteristics include cooperative work on ontologies, also broadcast and function reception.

1.5.3.Protégé [18].

Protégé was designed to build domain model ontologies. Developed by the Informatics Medic Section of Stanford, it assists software developers in the creation and support of explicit domain models and incorporation of such models directly into the software code. The methodology allows the system constructors to develop software from modular components to domain models and independent problem solving methods, which implement procedural strategies to accomplish tasks [19]. It is formed of three main sections: *ontology editor* (allows to develop the domain ontology by expanding the hierarchical structure, including classes and concrete or abstract slots); *KA tool* (can be adjusted to the user's needs by means of the "Layout" editor), and *Layout interpreter* (reads the output of the layout editor and shows the user the input-screen that is necessary to construct the examples of classes and sub-classes). The whole tool is graphic and beginner users oriented.

2. GeDA-3D Agent Architecture

GeDA-3D [20] is a final user oriented platform, which was developed for creating, designing and executing 3D dynamic virtual environments in a distributed manner. The platform offers facilities for managing the communication between software agents and mobility services. GeDA-3D Agents Architecture allows to reuse behaviors necessary to configure agents, which participate in the environments generated by the virtual environments declarative editor [20, 21]. Such agent architecture contains the following main features:

- *Skeleton animation engine*: This engine provides a completely autonomous animation of virtual creatures. It consists of a set of algorithms, which allow the skeleton to animate autonomously its members, producing more realistic animations, when the avatar achieves its objectives.
- *Goals specification*: The agent is able to receive a goal specification that contains a detailed definition of the way the goals of the agent must be

reached. The global behavior of the agent is goal-oriented; the agent tries to accomplish completely its specification.

- *Skills-Based Behaviors*: The agent can conform its global behavior by adding the necessary skills. It means that such skills are shared and implemented as mobile services registered in GeDA-3D.
- *Agent personality and emotion simulation*: Agent personality and emotion simulation make difference in behaviors of agents endowed with the same set of skills and primitive actions. Thus, the agents are entities in constant change, because their behavior is affected by their emotions.

Figure 1 presents the architecture of the platform GeDA-3D. The modules that constitute it are: *Virtual-Environment Editor* (*VEE*), *Rendering*, *GeDA-3D kernel*, *Agents Community* (*AC*) and *Context Descriptor* (*CD*). The striped rectangle encloses the main components of GeDA-3D: *scene-control* and *agents-control*.

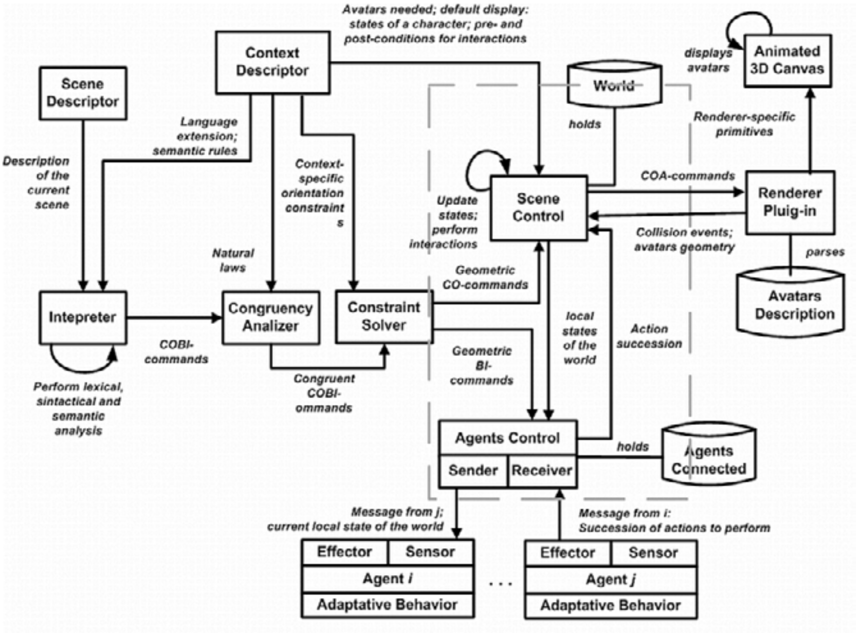

Avatars needed; default display: states of a character; pre- and post-conditions for interactions
Context Descriptor
Scene Descriptor
displays avatars
Animated 3D Canvas
World
holds
Renderer-specific primitives
Language extension; semantic rules
Description of the current scene
Context-specific orientation constraint s
COA-commands
Update states; perform interactions
Scene Control
Renderer Pluig-in
Collision events; avatars geometry
parses
Natural laws
Geometric CO-commands
Avatars Description
Intepreter
Congruency Analizer
Constraint Solver
local states of the world
Action succession
COBI-commands
Geometric BI-commands
Perform lexical, sintactical and semantic analysis
Congruent COBI-ommands
Agents Control
Sender
Receiver
holds
Agents Connected
Message from j; current local state of the world
Message from i: Succession of actions to perform
Effector
Sensor
Agent i
Adaptative Behavior
...
Effector
Sensor
Agent j
Adaptative Behavior

Figure 1. GeDA-3D Agent Architecture

The *VEE* includes the scene descriptor, interpreter, congruency analyzer, constraint solver, and scene editor. In fact, the *VEE* provides the interface between the platform and the user. It specifies the physical laws that govern the environment and describes a virtual scene taking place in this environment. The *Rendering* module addresses all the issues related to 3D graphics; it provides the design of virtual objects and the scene display. The *AC* is composed by the agents that are in charge of ruling virtual objects behavior.

The scene description provides an agent with detailed information about what task the user wants it to accomplish instead of how this task must be accomplished. Furthermore, the scene might involve a set of goals for a single agent, and it is not necessary that these goals are reached in a sequential way. The user doesn't need to establish the sequence of actions the agent must perform, it is enough to set goals and provide the agent with the set of primitive actions. The agents are able to add shared skills into their global behavior. So, the user describes a scene with the help of a high level language similar to human language. He or she specifies the goals to reach, that is to say, what the agent must do, not the way it must do it. It is important to highlight that one specification can produce different simulations.

3. Proposal

As described previously, the aim of our research is to provide the non-experienced final users with a simple and reliable tool to generate 3D worlds on the basis of description written in a human-like language. This article describes the *Virtual Environment Editor* of the GeDA-3D project. It receives the characteristics of the desired environment as input and validates all the instructions for the most accurate representation of the virtual world. The model includes the environment's context, the agents' instructions and the 3D view render modeling data. The novel approach uses the Knowledge Base [22] to obtain the information necessary to verify first the validity of the environment description, and later the validity of the model in the methodology process. Once this first step has been finished, the ontology is applied to obtain information required for visualizing the environment created in 3D and validating the actions computed by agents, taking in charge the behavior of avatars.

The main concern of this research is to assign the right meaning to each concept of the description introduced as input by the user. Human language is very ambiguous, so the parse of it is time and resource consuming. To avoid this hard work and make the writing of descriptions easy, we have defined a language completely oriented to scenario descriptions. We have called this language ***V***irtual ***E***nvironment ***De***scription ***L***anguage or VEDEL.

3.1. Virtual Environment Description Language (VEDEL)

VEDEL is a declarative scripting language in terms described in [23, 24]: A declarative language encourages focusing more on the problem than on the algorithms of its solution. In order to achieve this objective, it provides tools with high level of abstraction. A scripting language, also known as a *glue language*, assumes that there already exists a collection of components, written in other languages. It is not intended for creating applications from scratch. In our approach, this means that the user specifies what objects should be included in the scenario and describes their location, but he or she does not specify how the scenario must be constructed. It is also assumed that the required components are defined somewhere else, in our case in the object and avatar database. The definition of VEDEL using the EBNF (Extended Backus Normal Form) is:

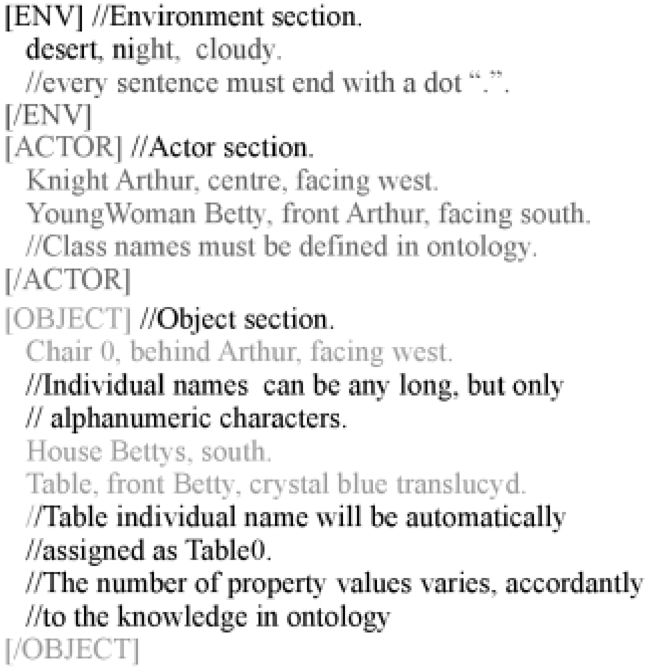
Alphabet:
Σ = {[**A** − **Z** | **a** − **z**] , [**0** − **9**] , , **.**}
Grammar:
Description := <environment> <actor> <object>
<environment> := **[ENV]** <sentence> **[/ENV]**
<actor> := **[ACTOR]** <sentence>* **[ACTOR]**
<object> := **[OBJECT]** <sentence>* **[/OBJECT]**
<sentence> := <class>(<**,**> <property>) ***.** >
<class> := <entity> (<identifier>)\
<property> := <propid> (<value>+)
Vocabulary:
<class> := <word>
<entity> := <word>, <entity> ∈ Ontology's Classes
<identifier> := <word>
<propid> := <word>, <propid> ∈ Ontology's Classes
<value> := <word> | <number> | <identifier>
<modifier> := <word>
< identifier >:= ([**A** ⊆←**Z** ♣←**a** ⊆←**z**] ♣←[**0** ⊆←**9**])+

< word >:= [**A** − **Z** | **a** − **z**]+
< number >:= [**0** − **9**] + (**.** [**0** − **9**]+)

The language is supposed to guide the user through the construction and edition of the description, separating the text into three paragraphs: Environment, Actors and Objects. Each paragraph is composed of sentences, at least one for the Environment paragraph, and any number for the Actors and Objects paragraphs. The sentences are composed of comma-separated statements, the first statement always being the entity's class, that is, a concept included in the Knowledge Base and an optional unique identifier. The rest of the sentences correspond to the features characteristic for a particular entity. Each sentence must end with a dot ".".

In the Environment section the user defines the context, the general setting of the scenario. The Actors section contains all the avatars, that is, entities with an autonomous behavior, which interact with other entities and the environment. It is important to notice that the Knowledge Base is the base for assigning behaviors to avatars. For instance, even when the entity "dog" is defined as an actor (autonomous entity), but the Knowledge Base does not define any features of its behavior (walk, bark, etc.), the actor will be managed as an inanimate object.

The description is validated through the lexical and semantic parser. This parser is basically a state machine, which prevents the entry for non-valid characters and bad formatted statements. If the parser finds some error in the description, the statement or the whole sentence is not taken in account for the modeling process, and an error message is produced to notify the user. The parser works with the following rule: given any description X, written under the VEDEL definition $G = \{\Sigma, N, P, S\}$ for the language L, X is a VEDEL compliant description, if and only if $X \subseteq L(G)$ and $L \Rightarrow_{i} X$.



```
[ENV] //Environment section.
  desert, night, cloudy.
  //every sentence must end with a dot ".".
[/ENV]
[ACTOR] //Actor section.
  Knight Arthur, centre, facing west.
  YoungWoman Betty, front Arthur, facing south.
  //Class names must be defined in ontology.
[/ACTOR]
[OBJECT] //Object section.
  Chair 0, behind Arthur, facing west.
  //Individual names  can be any long, but only
  // alphanumeric characters.
  House Bettys, south.
  Table, front Betty, crystal blue translucyd.
  //Table individual name will be automatically
  //assigned as Table0.
  //The number of property values varies, accordantly
  //to the knowledge in ontology
[/OBJECT]
```

Figure 2. Example of a description written in VEDEL

The parsed VEDEL entry is formatted as a hierarchical structure, so the modeler can access any sentence at any moment, usually making a single pass from the top entry (the environment) to the bottom entry. Each entry is in its turn formed of the class entry (the optional identifier, which is automatically generated in the incremental way, if the user doesn't explicitly establish one) and the properties of entries. We have decided to use the Protégé system due to its open API, written in Java, and also due to its simple, yet rich GUI, which allows quick creation, modification and maintaining of the Knowledge Base, easy to integrate into our project.

Once we have parsed the entry written in VEDEL, we need to establish the meaning for each concept introduced by the user. To accomplish this task, we rely on the Knowledge Base. This Knowledge Base stores all the information relevant for the concepts to be represented, from physical properties, such as size, weight or color, to internal properties, such as energy, emotions or stamina. The parsed VEDEL entry is revised by the inference function, which makes use of the ontology to validate the entities and their properties, as well as to validate the congruency in the scenario. This function also performs in the parsed entry the operations necessary to transform values from the string entry to primitive data form. The Knowledge Base consists of the following components:

- A finite set C of classes.
- A finite set P of properties.
- A finite set D = (C □ P) of class-properties assignments.
- If $x \in C$ and $y \in C$ is such that $y \in x$, y is a *subclass* of x.
- If $x \in P$ and $y \in P$ is such that $y \in x$, y is a *subclass* of x.

- Given a certain $D(x)$, $D(x) \supseteq D(y)$, for all y subclass of x.
- An *Object* property is such that $O = (C \lozenge C)$, where $\lozenge$ is the functional relationship (Functional, Inverse Functional, Transitive or Symmetric).
- A *DataType* property is such that $V = \{values\ \}$, where $\{values\}$ is a finite set of any typeset values.
- An Individual is the instantiation of the class x, in such form that every p in $P(x)$ is assigned with a specific set (empty or otherwise).

In our project, $C = \{Environment,\ Actor,\ Object,\ Keywords\}$, and the following subsets of properties are obligatory for each subclass of C elements:

- For all $x \in Environment$, $Q_e = \{size,\ attributes\}$, $Q_e(x) \subseteq D(x)$.
- For all $x \in Actor$, *Objects*, $Q_a = Qe \cup \{geozones,\ geotags,\ property\_type\}$, $Q_a(x) \subseteq D(x)$.
- For all $x \in Keywords$, $Q_k = Q_a \cup \{attributes\}$, $Q_k(x) \subseteq D(x)$.

The ontology has been described with respect for naming conventions. However, in addition any class must have at least one individual named as the class **with** the "_default" suffix at the end. This is the most basic representation for the class. Figure 3 shows the main structure of the ontology, the properties currently added and the individuals view. Any additional class will only be used by the output generation function, but the relationships between the subclasses and the rest of the ontology will be respected by the environment. For instance, Figure 3 shows additional classes “laws”, “actions” and “collisions”. While these classes are not required directly by the final user by means of calling their members, the modeler will take into account any relationship between the main four classes and these additional classes. This means that the object property “AllowedOn” will be used to validate the actions that can be performed in a given environment not at the modeler's level, but as an explicit rule written for the context and the underlying architecture, since only the actions linked to the environment can be assigned to the architecture.

The process of validating any expression found in the description is carried out by the inference function. This function accesses the Knowledge Base, searching for the individual that matches the pattern “*<expression>_default*". We can call it the *definition individual*. This individual can be the obligatory class definition or some other individual. If there is no definition individual for the expression, an error condition will arise, and the statement or the sentence will be excluded from the rest of the modeling process. If the definition individual is found, the inference function precedes to gather all the information stored in the Knowledge Base for that expression, according to the parameter it has received. In the case of entities definitions, it subtracts all the basic information, such as size, specialized tags and the properties specified in the *attribute* property. For the entity's properties, the inference function searches for the corresponding definition individual and gathers the data according to the values stored in the *property type* property. This differentiation is made since there are cases, when the object can be a standalone entity or part of another, bigger entity. Also, the same concept can be used for different requests, i.e., the concept "north" can be used to either place an entity in a specific position, or to indicate the orientation of the entity. The rules for the inference function are:

$$\forall x \in \{description\}, \exists y \in C \vee y \in I \Rightarrow P(y) = P(x) \vee P_n(y) = x$$

$$\text{If } P(x) = P(y) \Rightarrow x, y \in C \wedge \exists x_i, y_i \in I$$

$$\text{If } P_n(y) = x \Rightarrow y \in C \wedge ((x \in D \wedge x_i \in I) \vee x \in V)$$

$$\forall x \in C \Rightarrow \exists x_i \in I$$

$$\forall y \in C \Rightarrow \exists y_i \in I$$

The gathered information is formatted as the basic entry for the model data structure, which is a hierarchical structure with the entity entries as the top levels and the properties for each one of the entities as leaves, as depicted in Figure 4. Each entry is instantiated with the basic definition: the feature values for successful representation of the entity in the model, according to the knowledge stored in the Knowledge Base. These basic entries are then modified with the help of values obtained from the parsed entry description, once they have passed the inference process, have been validated and transformed to the format suitable for the underlying architecture.

3.2.Knowledge in algorithms for constraint satisfaction problems

In [25], Hunter defines the Constraint Satisfaction Problems as follows: a CSP is a tuple composed by a finite set of variables *N*, a set of possible values *D* for the variables in *N* and a set of constraints *C*. For each variable N_i, there is a domain D_i, so that to each variable can only be assigned values from its own domain. A restriction is a subset of the variables in N and the possible values that can be assigned them from D. Constraints are named according to number of variables they control. A solution to a CSP is an assignment of values form the set of domains D to each variable in the set N, in such way that none of the constraints is violated. Each problem that presents solution is considered satisfied or consistent; otherwise it is called non-satisfied or inconsistent.

To solve a CSP, two approximations can be used: *search* and *deduction*. The search generally consists in selecting an action to develop, maybe with the aid of a heuristic, which will lead to the desired objective. Tracking is an example of search for CSP; this operation is applied to value assignation for one or more variables. If no value able to keep the consistency of the solution can be assigned to the variable (the dead-end is reached), the tracking is executed.

There are several methods and heuristics for solving CSPs, among them backtracking [26], backmarking [27], backjumping [28], backjumping based on graphics [29] and forward checking [30].

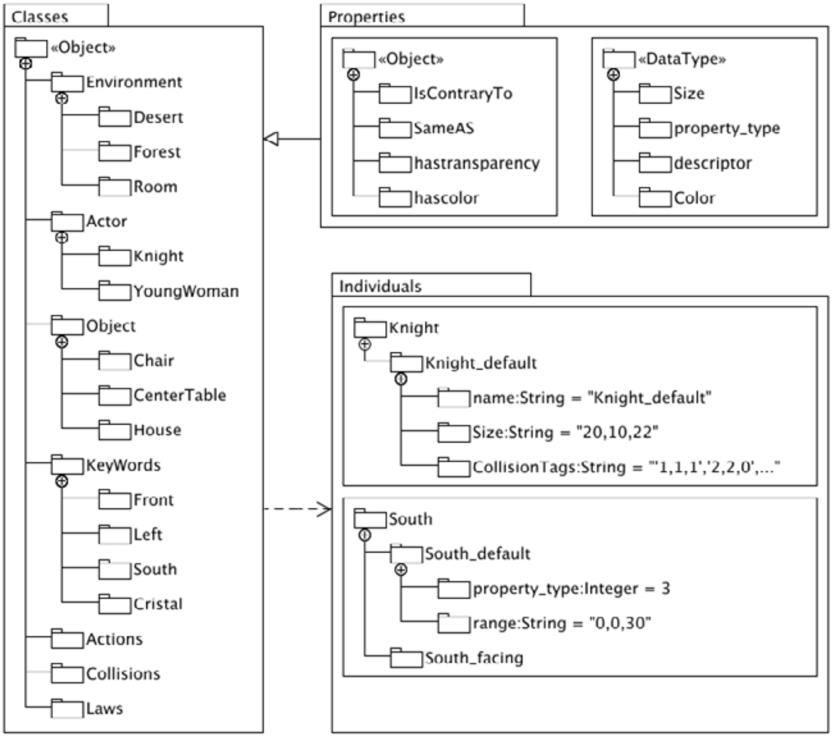


Figure 3. The ontology used by the Virtual Editor

The parsed entry obtained from the description and validated through the inference function is sent to the modeling function, which generates the first model that is called zero-state model. At this stage, all geometrical values are set to default: either to zero or to the entity's default values and the corresponding request is stored in the model for quick access and verification. Objects whose position is not specified are placed in the center of the scenario (geometrical origin) and the specific positions are respected. For those entities whose position is established in relation to others, a previous evaluation is conducted in order to position first the entities referenced. Any specified position is assigned, even if there are possible conflicts with other elements, the environment, the rules of the environment or the properties of the entity.

With the objects in their initial positions and postures, the model is sent to the logical and geometrical congruency analyzer. First it is verified, if no properties are violated with the current requests (objects with postures that cannot be represented, objects positioned over entities that cannot support them, etc.), and then the geometry of the scenario is verified for spatial conflicts. The validation of the current model state, as well as the modifications made to obtain a valid model state that satisfy the user request, is achieved by means of the Constraint Satisfaction Problem (CPS) solving algorithm, whose parameters are detailed as follows:

A set $V = \{x_1, \ldots, x_n\}$ of variables where $x = \{P, O, S\}$ such as

- $P = \{x, y, z\}$ is the entity's position in the environment.
- $O = \{\theta_x, \theta_y, \theta_z\}$ is the entity's orientation.
- $S = \{S_x, S_y, S_z\}$ is the entity's size, including the scale.

A domain D for the set V such as

- $D(P) = \Re^3$
- $D(O) = [0, 2\pi], \forall \theta \in O$
- $D(S) = \Re^3$.

A set C of constraints, formed by the next functions:

$$(1.a) \quad (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - (r_1 + r_2) = 0$$

$$(1.b) \quad \left(\frac{x}{p}\right)^2 + \left(\frac{y}{q}\right)^2 - 2z = 0$$

$$(1.c) \quad \left(\frac{x}{dx}\right)^2 + \left(\frac{y}{dy}\right)^2 + \left(\frac{z}{dz}\right)^2 - 1 = 0$$

Before validating the request, information for each spatial property is obtained from the Knowledge Base in the form of vectors that indicate positioning, either absolute or relative, the maximum distance to which the objects can be separated from each other still fulfilling the requests, and orientation, relative or absolute.

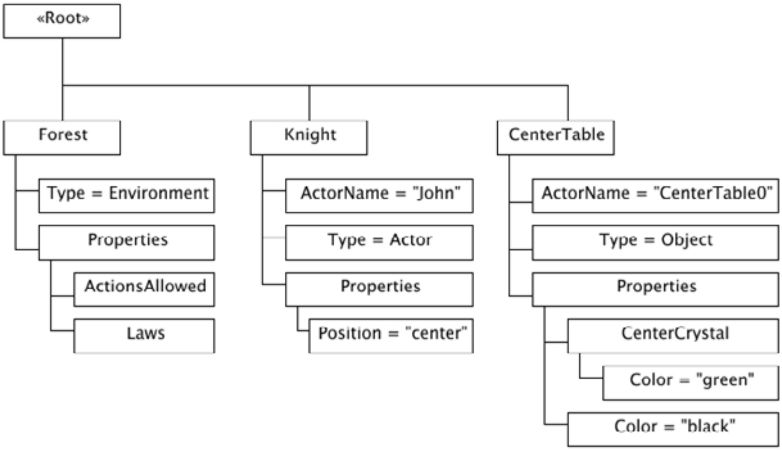


Figure 4. Data structure obtained from a Description

Once position and orientation have been set for each entity, the geometrical analyzer verifies that no collision occurs between objects. This is achieved with the help of several collision tags also stored in the Knowledge Base for each entity. These tags are extracted, instantiated with the entity's parameters, and then compared with the others entities' tags. This is our first constraint, since there must be no collision between tags to declare the model valid. The only exceptions to this rule are the indications for close proximity between the entities (zero distance or distance against request). The tags are modeled as spheres that cover all or most of the entity's geometry (see figure 5). The equation for verifying the collision between two spheres is simple and quick to solve (see equation 1.a in the CSP parameters definition).

While collision consistency is being conducted, other tests are also conducted to secure position consistency. Each positional statement (left, right, front, behind) has a valid position volume assigned, where the objects that hold a spatial relationship to the entity must be put, once this volume has been instantiated. The test function is a quadratic equation, represented in the model as a parabolic (see equation 1.b in the CSP parameters definition). This representation was chosen, because it is relatively easy to solve, allows the verification of spaces nearly of the same size as the classic collision boxes and can be shaped to the entity's measures. If the entity being positioned does not pass the consistency test, that is, none of its characteristic points (figure 6) is inside the consistency function, a new position is computed, and the collision and positioning revision is executed again. In the specific case where two entities have the same relation to a third one, a new position is computed and the characteristic points tested for each entity. When it is not possible to find a new position that meets both constraints, a model error is generated and the user is informed about it.

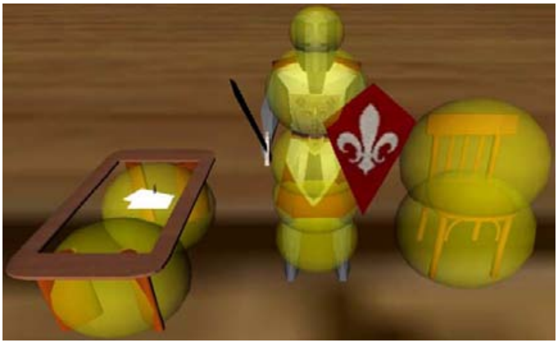


Figure 5. Collision tags examples

For special positioning request there is a third type of consistency function, the ellipsoid (see equation 1.c in the CSP parameters definition). These special positions are the inside, over, under, and against concepts. The reason for the choice of this type of equations is the capacity to shape the volume in different sizes: the volume that covers all or the majority of the space inside or as part of the entity, or a smaller volume set in the corresponding area of the target entity. The same heuristics are applied to the other position request. World boundaries are validated through this function. For the entities in the Actors section, requests are sent to the animation module, which sets the position of each articulation in the entity to obtain the desired posture. This information is used to provide the corresponding collision tags and characteristic points, which in turn allow the modeler to verify consistency for the postures and actions the actor will be performing. If several solutions are found for the current request, the user can choose the one that fits better the requests. Normally, a well constrained problem would find a finite number of solutions, but in the case of under-constrained problems, it could be an infinite number of solutions. In these cases, the modeler can show only a predefined number of solutions, or can show in iterative form all the possible solutions, until the user chooses one.

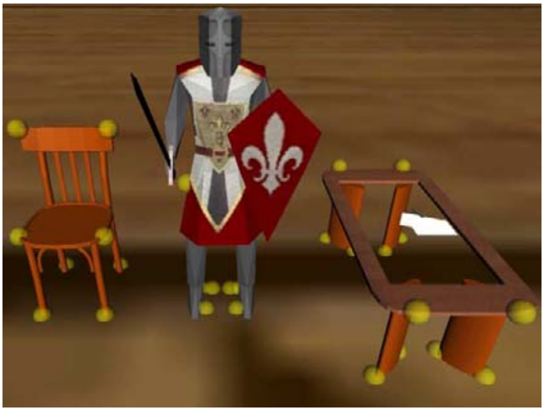


Figure 6. Characteristic points for different entities

The final step in the modeling process is the generation of the necessary outputs in order to allow the created environment to be visualized and modified. These outputs are generated using the MVC or Model View Controller. This method provides any desired modifications in the outputs that will be generated without modifying the code for the modeler, and allows the users to customize the outputs and integrate the modeler in multiple applications.

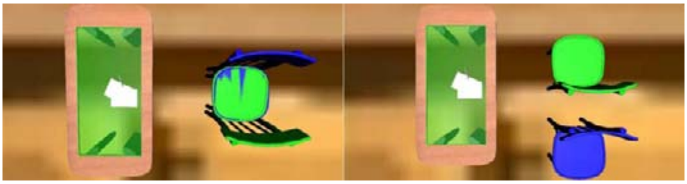


Figure 7. Conflict solving example

4. Results

Until now the parser for the VEDEL language has been implemented and it was functionally working. The necessary links with the GeDa-3D architecture were established in order to obtain a visual representation of the descriptions written in this language.

4.1.The VEDEL Parser and Modeler

The parser was created in Java language to reach multiplatform capabilities and compatibility with the rest of the GeDA-3D architecture, using the JDK 1.5.0_07-b03 [31]. Our parser is basically a state machine: each section of the description corresponds to a state, as well as each sentence and each property. If the state generates an error output, the process is stopped and an error condition arises. Each word is considered to be a token and validated by the inference machine, with the exception of numbers, particular identifiers and closing/opening constructions. The inference machine and the modeler described in the previous section are used to validate semantically the parsed description, and then the data structure that represents the model is obtained. Finally, the outputs are generated using the MVC (Model View Controller) function. The templates are formatted, so they can be filled with the data stored in the data structure, and the formatted output of each entry in the model forms the complete output.

4.2.The GeDA-3D prototypes

To obtain a visual output of the description written in VEDEL, the prototype of the GeDA-3D architecture was created. The prototype has a kernel [32], a render working upon the AVE (Animation of Virtual Creatures) project [33], and our parser. This prototype works as follows: the kernel received the outputs generated by the parser (one output for the kernel and one in LIA-3D, Language of Interface for Animations in 3D presented in [33], for the parser). Then it generates the necessary agents, and the AVE output is sent to the render module, where the scenario is composed, rendered and presented to the final user. Next, we present some examples obtained by the X3D [34] based output (figures 8 to 10) and their corresponding descriptions.

Description 1.

[ENV]
forest.
[/ENV]

[ACTOR]
Knight A, center.
Knight B, left A.
Knight C, right A.
Knight D, behind B.
Knight E, behind C.
[/ACTOR]

[OBJECT]
House, behind (80) A.
[/OBJECT]

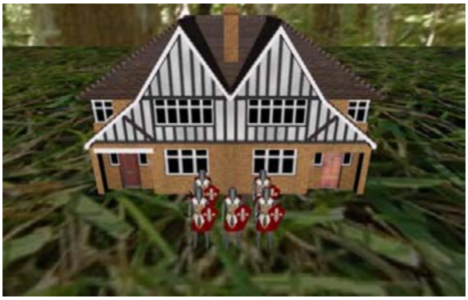


Figure 8. Result obtained for description 1

Description 2.

[ENV]
theater.
[/ENV]

[ACTOR]
Knight A, center.
Knight B, left A.
Knight C, right A.
YoungWoman D, behind B, facing A.
YoungWoman E, behind C, facing A.
[/ACTOR]

[OBJECT]
[/OBJECT]

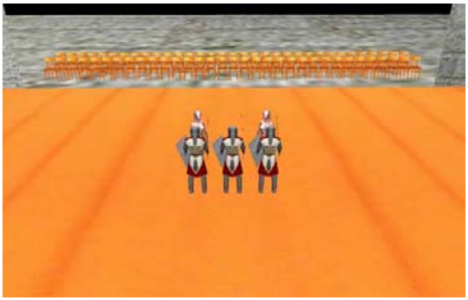


Figure 9. Result obtained for description 2

Description 3.

[ENV]
void.
[/ENV]

[ACTOR]
[/ACTOR]

[OBJECT]
Chair, color blue.
Chair, color red.
Chair, color green.
[/OBJECT]

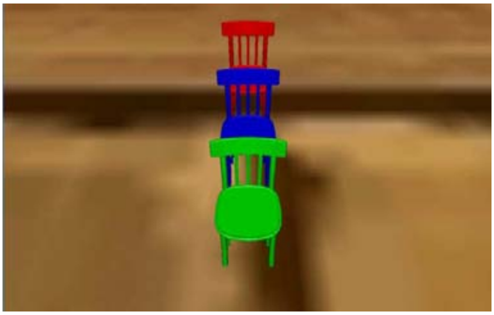


Figure 10. Result obtained for description 3

5. Conclusions

Our contribution to the research topic described in this article concerns declarative modeling for creation of scenarios. This problem is important, because the design of virtual scenarios is time- and labor-consuming even for expert users who have appropriate tools at their disposal. In this article we contribute mainly to the use of knowledge databases to support and accelerate the semantic analysis of sentences that compose the declarative form of a scenario. More specifically, our proposal uses a Knowledge Base in the three phases that constitute the declarative modeling. During the *Description* phase the Knowledge Base helps to get semantic elements necessary for verifying the description, that is, the input to the system. For the *Model Creation* the modeler will use the Knowledge Database to obtain the information necessary for the model creation. To accomplish this task the modeler uses a restriction satisfaction algorithm supported by the Knowledge Base. Finally, the user applies the Knowledge Base in the *Vista* phase to obtain information about the requirements which could not be satisfied, in case there wasn't found any solution, or select one of the possible solutions.

The approach we have proposed shows two very important advantages: The first one concern the fact that the solution obtained in this way can be used in systems able to evolve a scene, as the GeDA-3D project described in this article. The second one concerns the possibility of expanding the available environments and entities just by increasing the knowledge database, leading the declarative editor towards a generic, open architecture.

Some of the results obtained include: a structured method for creating and editing descriptions with the use of tools that validate the inputs, such as lexical and semantic analyzers; the implementation of prototypes useful for visualizing the generated scenario on the basis of the respective description. These results are important because they validate our proposal. This validation proves the possibility for creating more types of scenarios, just increasing the knowledge database with the corresponding information.

Our future work includes the development of more robust CPS algorithms supported by the Knowledge Base, that not only consider the physical properties of the entities, but also the context properties and the semantic properties; a semantic validation function, which verifies, if every entity inserted in the environment is capable or allowed to exist in such environment, and, in some cases, provides the necessary changes in the entity's properties, so it can get a valid element; and finally, the complete integration with the GeDA-3D architecture.

References

[1] K. Victor. Flux Studio Web 3D Authoring Tool. http://wiki.mediamachines.com/index.php/Flux_Studio, 2007.

[2] Last Software. Google Sketchup. http://sketchup.google.com/, 2008.

[3] Autodesk. 3DS MAX 9 Tutorials. http://usa.autodesk.com/adskservlet/item?siteID=123112&id=8177537, 2007.

[4] Autodesk. Autodesk Maya Help. http://www.autodesk.com/us/maya/docs/Maya85/wwhelp/wwhimpl/js/html/wwhelp.htm, 2007.

[5] J. S. Monzani, A. Caicedo and D. Thalmann. Integrating Behavioral Animation Techniques. In EG 2001 Proceedings, volume 20(3), pages 309–318. Blackwell Publishing, 2001.

[6] D. Plemenos, G. Miaoulis and N. Vassilas. Machine Learning for a General Purpose Declarative Scene Modeller. In International Conference GraphiCon'2002, Nizhny Novgorod (Russia), September 15-21, 2002.

[7] V. Gaildrat. Declarative Modelling of Virtual Environment, Overview of Issues and Applications. In International Conference on Computer Graphics and Artificial Intelligence (3IA), Athenes, Grece, volume 10, pages 5–15. Laboratoire XLIM - Université de Limoges, may 2007.

[8] J.-E. Marvie, J. Perret and K. Bouatouch. The FL-System: A Functional L-System for Procedural Geometric Modeling. The Visual Computer, pages 329 – 339, June 2005.

[9] Y. I. H. Parish and P. Müeller. Procedural Modeling of Cities. In SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pages 301–308, New York, NY, USA, 2001. ACM Press.

[10] B. Coyne and R. Sproat. Wordseye: An Automatic Text-To-Scene Conversion System. In SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pages 487–496. AT&T Labs Research, 2001.

[11] G. Kwaiter, V. Gaildrat and R. Caubet. Dem2ons: A High Level Declarative Modeler for 3D Graphics Applications. In Proceedings of the International Conference on Imaging Science Systems and Technology, CISST'97, pages 149–154, 1997.

[12] W. Ruchaud and D. Plemeno. Multiformes: A Declarative Modeller as a 3D Scene Sketching Tool. In ICCVG, 2002.

[13] K. Xu, A. J. Stewart and E. Fiume. Constraint-Based Automatic Placement for Scene Composition. In Graphics Interface, pages 25–34, May 2002.

[14] A. Farquhar, R. Fikes and J. Rice. The Ontolingua Server: A Tool for Collaborative Ontology Construction. Technical report, Knowledge Systems Laboratory, Stanford University, 1996.

[15] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2):199–220, June 1993.

[16] J. Domingue. Tadzebao and Webonto: Discussing, Browsing, and Editing Ontologies on the Web. In Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, April 1998

[17] E. Motta. Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1999.

[18] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu and M. A. Musen. Knowledge Modeling at the Millennium (the Design and Evolution of Protégé-2000). Technical Report, Stanford Medical Informatics, 1998.

[19] H. Eriksson, Y. Shahar, S. W. Tu, A. R. Puerta and M. A. Musen. Task Modeling with Reusable Problem-Solving Methods. Artificial Intelligence, 79(2):293–326, 1995.

[20] F. Zúñiga, F. F. Ramos and I. Piza. GeDA-3D Agent Architecture. Proceedings of the 11th International Conference on Parallel and Distributed Systems, pages 201–205, Fukuoka, Japan, 2005.

[21] H. I. Piza, F. Zúñiga and F. F. Ramos. A Platform to Design and Run Dynamic Virtual Environments. Proceedings of the 2004 International Conference on Cyberworlds, pp. 78-85, 2004.

[22] J. A. Zaragoza Rios. Representation and Exploitation of Knowledge for the Description Phase in Declarative Modeling of Virtual Environments. Master’s thesis, Centro de Investigación y de Estudios Avanzados del I.P.N., Unidad Guadalajara, 2006.

[23] C. E. Chronaki. Parallelism in Declarative Languages. PhD thesis, Rochester Institute of Technology, 1990.

[24] J. K. Ousterhout. Scripting: Higher Level Programming for the 21st Century. IEEE Computer Magazine, 31(3):23–30, March 1998.

[25] D. H. Frost. Algorithms and Heuristics for Constraint Satisfaction Problems. PhD thesis, University of California, 1997. Chair-Rina Dechter.

[26] S. W. Golomb and L. D. Baumert. Backtrack Programming. J. ACM, 12(4):516–524, 1965.

[27] S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Pearson Education, 2003.

[28] J. G. Gaschnig. Performance Measurement and Analysis of Certain Search Algorithms. PhD thesis, Carnegie-Mellon Univ. Pittsburgh Pa. Dept. Of Computer Science, 1979.

[29] R. Dechter. Enhancement Schemes for Constraint Processing: Backjumping, Learning, and Cut Set Decomposition. Artificial Intelligence, 41(3):273–312, 1990.

[30] R. M. Haralick and G. L. Elliott. Increasing Tree Search Efficiency for Constraint Satisfaction Problems. Artificial Intelligence, 14(3):263–313, 1980.

[31] SUN Microsystems. The Java SE Development Kit (JDK). http://java.sun.com/javase/downloads/index.j sp, 2007. Last visited 02/01/2007.

[32] A. G. Aguirre. Nucleo GeDA-3D. Master's thesis, Centro de Investigación y de Estudios Avanzados del I.P.N., Unidad Guadalajara, 2007.

[33] A. V. Martínez González. Lenguaje para Animación de Criaturas Virtuales. Master's thesis, Centro de Investigación y de Estudios Avanzados del I.P.N., Unidad Guadalajara, 2005.

[34] X3D. The Virtual Reality Modeling Language - International Standard ISO/IEC. http://www.web3d.org/x3d/specifications/, 2007. Last visited 06/01/2007.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-133

Further Results on Multiobjective Evolutionary Search for One-Dimensional, Density Classifier, Cellular Automata, and Strategy Analysis of the Rules

Gina M.B. OLIVEIRA [a], José C. BORTOT [b] and Pedro P.B. de OLIVEIRA [c]
[a] *Universidade Federal de Uberlândia, Faculdade de Computação*
Av. João Naves de Ávila 2160, Bloco B, Campus Santa Mônica; 38400-902
Uberlândia , MG, Brazil, gina@facom.ufu.br
[b] *Fundação Armando Álvares Penteado, Faculdade de Computação e Informática*
[c] *Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informática &*
Pós-Graduação em Engenharia Elétrica

Abstract. A strong motivation for studying cellular automata (CA) is their ability to perform computations. However, the understanding of how these computations are carried out is still extremely vague, which makes the inverse problem of automatically designing CA rules with a predefined computational ability a fledgeling engineering endeavour. Various studies have been undertaken on methods to make CA design possible, one of them being the use of evolutionary computational techniques for searching the space of possible CA rules. A widely studied CA task is the Density Classification Task (DCT). For this and other tasks, it has recently been shown that the use of a heuristic guided by parameters that estimate the dynamic behaviour of CA can improve a standard evolutionary design. Considering the successful application of evolutionary multiobjective optimisation to several kinds of inverse problems, here one such technique known as Non-Dominated Sorting Genetic Algorithm is combined with the parameter-based heuristic, in the design of DCT rules. This is carried out in various alternative ways, yielding evolutionary searches with various numbers of objectives, of distinct qualities. With this exploration, it is shown that the resulting design scheme can effectively improve the search efficacy and obtain rules that solve the DCT with sophisticated strategies.

Keywords. Cellular automata, evolutionary multiobjective optimization, density classification task, non-dominated sorting genetic algorithm (NSGA), parameter-based forecast of dynamic behaviour, inverse design.

Introduction

The relationship between the dynamic behaviour of cellular automata (CA) and their computational ability has been a recurring theme in complex systems research [1], [2]. Evolutionary methods have been used in the inverse problem of designing CA with predefined computational abilities. A widely studied CA task having been the ability to solve the density classification task (DCT), and several evolutionary computation techniques have been used to look for CA rules of such a kind [3], [4], [5], [6], [7], [8].

Although the DCT itself has no apparent practical interest, it is a paradigmatic example of a problem that requires global coordination to be solved, thus posing a challenge to its solution by means of cellular automata. Even for evolutionary techniques, the huge size of CA rule spaces is a serious obstacle that turns the search slow and many times non-effective.

Aiming at the reduction of the latter problem, we previously used a set of static parameters (that is, directly derived from the CA rule table) as a heuristic to guide the processes underlying the genetic search in the space of possible designs [7]. Based on some indicators, which estimate the dynamic behavior of a cellular automaton rule, it was then possible to build a guide to a standard genetic algorithm (GA) [9] to find CA rules for the Density Classification Task.

In this approach the parameter-based heuristic was incorporated into the GA in its associated fitness function, and in the action of the genetic operators (of crossover and mutation). The fitness function of every candidate cellular automaton rule was then worked out by the weighed sum between a fitness component due to its efficacy in the DCT on a sample of initial configurations (ICs), and a second fitness component that represents the bias due to the parameter-based heuristic [7]. Although this solution yielded an improvement on the genetic search in all the tasks studied, the weight of the heuristic in the rule evaluation was observed to interfere on this performance.

Instead of using any weighed sum to evaluate the rule, experiments are reported here where an evolutionary multiobjective approach is used. Accordingly, the heuristic and the efficacy in the IC sample are kept separate, as independent objectives to be followed by the genetic search.

Although we have shown in previous works that the parameter-based heuristic could effectively improve evolutionary design of cellular automata rules [7], [10], its incorporation in the algorithm was carried out in a very straightforward way. The whole idea of going from that earlier usage of the heuristic to the one discussed herein is the quest for a better way for the parameter-based heuristic to be accounted for. What the results in the last sections of this paper show is that the multiobjective approach can in fact enhance the heuristic's performance, which is reflected in the algorithm designing CA rules with more sophisticated strategies and, consequently, higher efficacies.

The work we discuss herein is emblematic as an heuristic-oriented approach to a problem not only because of the nature itself of our approach to the DCT, but also because, after many years of research efforts reported in the literature, targeting the solution of the problem, it was then proven to be impossible to solve the DCT perfectly, by any one-dimensional cellular automaton with finite radius and periodic boundary conditions [11] (but see also [12], for a subsequent view on the same issue); in other words, there will always be classification failures in the space of possible initial configurations. As a consequence, since no good solution to the DCT can solve it exactly, this naturally makes it an interesting candidate for a heuristic approach.

It is worth pointing out, though, that perfect solutions can be given to alternative formulations of the task, such as by allowing the cellular automaton to have non-periodic boundary conditions [13], by allowing the application of two distinct elementary cellular automata rules in sequence [14], or by changing the classification criterion [15].

All in all, up to this date, the best possible imperfect rule for the DCT remains unknown; and this is what ultimately drives us. The trend to look for better and better DCT rules has led to better and better algorithms, more and more fine-tuned to the problem. As a consequence, as knowledge progresses towards better algorithms, to some extent it may be possible to transfer knowledge acquired in the DCT to similar problems. After all, it is reasonable to expect that the structure of the fitness landscape associated with the DCT preserves some resemblance to the fitness landscapes of other CA-based problems of a similar kind, namely, one-dimensional CA problems of binary nature and requiring global coordination to be solved. Naturally, such an issue would be extremely beneficial, since the possibility of devising CA rules able to perform other computational tasks would help our understanding of how CA are able to compute.

Our concern in the present paper is, therefore, less on trying to develop a new evolutionary multi-objective algorithm that can cope with the problem, and more on finding ways that can make it work in an effective way, in the specific problem instance represented by the DCT.

In the next section multiobjective optimization is characterized, and the particular multiobjective evolutionary technique our genetic search is based on – namely, NSGA – is briefly described. CA basic concepts are presented in sequence. Then, the basic elements related to the heuristic search for CA we have developed in previous works are reviewed. In the subsequent section the DCT is discussed, in connection with the heuristic information we have used in the context of standard evolutionary search for CA that might solve it. Finally, the experiments performed are described and the results discussed.

1. Evolutionary Multiobjective Methods

Many real world problems involve simultaneous optimization of multiple objectives, so that it is not always possible to achieve an optimum solution in respect to all of them individually considered. In this kind of problem, there is a set of solutions better than all the other solutions in the search space [16], which is named the *Pareto* front or set of non-dominated solutions.

Suppose that there are N objectives $f_1, f_2, \dots f_N$ to be simultaneously optimized. A solution A is said to be dominated by another solution B, or B dominates A, if B is better than A in relation to at least one of the objectives f_i, and is better than or equal to A in relation to all other objectives $f_1, \dots f_{i-1}, f_{i+1} \dots f_N$. Two solutions A and B are non-dominated in relation to each other if A does not dominate B and B does not dominate A. For example, suppose that functions f_1 and f_2 in Figure 1 must be simultaneously maximized. One can affirm that solution A is better than solutions C and D; that is, C and D are dominated by A. However, in the case of solutions A and B, it is not possible to affirm which one is the best. Therefore, one can say that solutions A and B are non-dominated and they dominate solutions C and D.

The Pareto front is the set of non-dominated solutions considering the entire search space, that is, any candidate solution that is better than one of the solutions from the Pareto front in respect to one of the objectives, is guaranteed to be worse with respect to another objective.

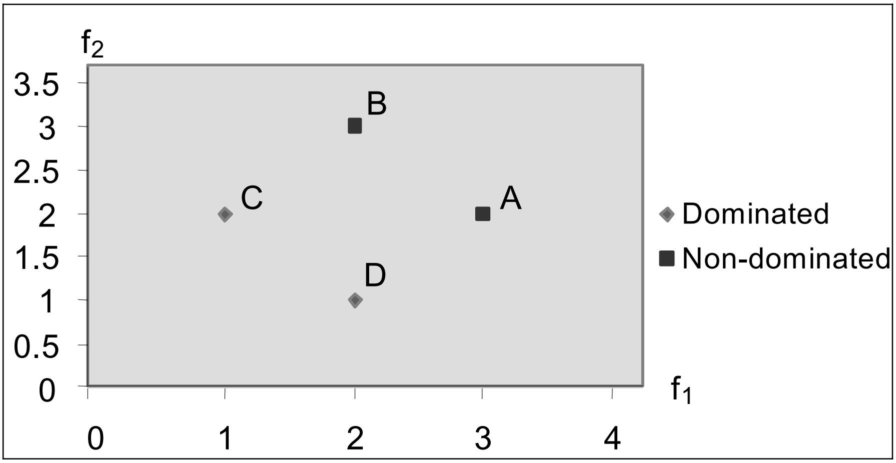


Figure 1. Example of dominated and non-dominated solutions when maximizing f_1 and f_2.

Multiobjective evolutionary methods try to find this solution set by using each objective separately, without aggregating them as an unique objective. Some of the most traditional multiobjective evolutionary methods are VEGA (Vector Evaluated Genetic Algorithm; Shaffer, 1985 [17]), NSGA (Nondominated Sorting Genetic Algorithm; Srinivas and Deb, 1994 [18]), MOGA (Multiple Objective Genetic Algorithm; Fonseca and Fleming, 1993 [19]), and NPGA (Niched Pareto Genetic Algorithm; Horn and Nafpliotis, 1993 [20]); the NSGA is briefly described below, insofar as it provided the procedures used in the present work.

More recently, new evolutionary methods have been developed, characterised by the fact that – unlike the methods mentioned above – an elitist strategy was incorporated in them, leading to methods such as NSGAII (Deb, Agrawal, Pratab and Meyarivan., 2000 [21]), SPEA (Zitzler and Thiele, 1999 [22]), SPEA2 (Zitzler, Laumanns and Thiele, 2001 [23]), PAES (Knowles and Corne, 1999 [24]) and PESA (Knowles, Corne and Oates, 2000 [25]). For a thorough account on the most important multi-objective evolutionary algorithms, and a comprehensive collection of applications, including innumerous examples in science and engineering, the reader should refer to [26], [27], [28] and [29].

The Nondominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb (1994) and is based on the concept of non-domination [18]. The basic difference of NSGA in relation to a simple GA is the way in which the individuals are evaluated. Basically, in order to obtain the fitness value of an individual, instead of using the fitness components associated with each objective involved in the problem, these components are used to rank the individuals according to their degree of domination over the others in the population; it is this measure of domination that is used as the fitness value that guides the action of the genetic operators and the selection process.

In order to work out that ranking, the population is organized in several layers of non-domination, the outermost layer representing the less dominated individuals, and the innermost containing the most dominated ones. Initially, all individuals in the population that are non-dominated are separated as the first, outermost layer, and a (dummy) fitness value is assigned to them, whose role is simply to characterize their degree of domination over the others in the population. The highest fitness value is assigned to the outermost layer, as the individuals in it exhibit the highest degree of domination (the actual value assigned is proportional to the population size; this and other details are being omitted here for brevity).

Then, the remaining individuals are classified again, also based on the non-domination criterion, and the second layer is formed with a (dummy) fitness lower than the first one. This process continues until all individuals are classified in their respective layers.

Once the non-domination layers were obtained, in order to maintain the diversity of the individuals inside each layer, a sharing function method is used to assign fitness to each individual [9]. The basic idea of the sharing function is to reduce the fitness of the individuals that have many neighbours in the same layer. The fitness reduction is by a factor that depends on the number and proximity of neighbouring individuals, so that, the larger the number of individuals lying close to each other, the higher their fitness reduction; analogously, the more isolated an individual, the smallest the influence of the fitness reduction. As a consequence, the sharing function induces the formation of niches, thus increasing diversity in the population.

After the non-domination classification and fitness assignment are performed, a proportional selection scheme is used so that the outer the layer an individual is in, the likelier its chance to reproduce. Stochastic remainder selection was the actual scheme used, preserved here because it was present in the original proposition of the NSGA; in this method, the next generation is formed by taking as many copies of every individual in the current population as the result of the integer division of its fitness value by the average fitness of the population; then, if the population needs be filled in, the fractional part of the latter division is used.

The other steps of the NSGA are very similar to the simple GA [9] and, as mentioned before, no elitist strategy is used, that is, no individual is allowed to bypass the selection process above, directly passing on to the next generation, regardless of how well fitted it may already be.

The fact that we centred our multiobjective approach around a non-elitist method such as NSGA, instead of using a more recent, elitist approach, such as NSGA-II [21], derives from the problem being tackled. Indeed, in the context of evolving CA to perform the DCT, the standard genetic search presented in [3] has been used to compare different approaches to the problem [4], [5], [6], [7]. In tune with that framework – that used a single objective, and a very elitist strategy for the selection criteria both for crossover and reinsertion of individuals in the population – it was a natural decision in the construction of the present multiobjective experiment to maintain all the basic structure of that reference GA. Therefore, because the main loop of our algorithm preserves the original experiment in [3], our resulting multiobjective algorithm is, indeed, elitist.

Finally, the fact that we used NSGA, instead of NSGA-II, which is computationally faster, was due to a technicality, that is irrelevant to discuss for present purposes. As will be shown in Section 4, this approach was sufficient to yield good results; but naturally, it is still possible to evaluate other multiobjective approaches as further investigations.

2. Cellular Automata

Cellular automata (CA) are discrete complex systems that possess both a dynamic and a computational nature. For what follows, the notation adopted is drawn from [1].

Basically, a cellular automaton consists of two parts: the cellular space and the transition rule.

Cellular space is a regular lattice of N cells, each one with an identical pattern of local connections to other cells, and subjected to some boundary conditions. The set of states in a cell is denoted by Σ and the number of states in the set by k. Each cell is referred to by an index i, and its state at time t by S_i^t, where $S_i^t \in \Sigma$. State S_i^t of cell i, together with the states of the cells connected to cell i, is named the neighbourhood η_i^t of cell i.

The transition rule, represented by $\Phi(\eta_i^t)$, yields the next state for each cell i, as a function of η_i^t. At each time step, all cells synchronously update their states according to $\Phi(\eta_i^t)$.

In computational terms, a cellular automaton is, therefore, an array of finite automata, where the state of each automaton depends on the state of its neighbours.

For one-dimensional CA, the size m of the neighbourhood is usually written as $m = 2r + 1$, where r is called the *radius* of the automaton. In the case of binary-state CA, the transition rule is given by a state transition table which lists each possible neighbourhood together with its output bit, that is, the updated value for the state of the central cell in the neighbourhood. The lattice is subjected to boundary conditions. These conditions are usually periodic, in which the lattice is like a ring.

Figure 2 displays an example of one-dimensional binary cellular automaton defined over a lattice of eleven cells. The neighbourhood of each cell consists of the cell itself and its two nearest neighbours. It is also known as radius 1 neighbourhood. The figure highlights the updating of the eighth cell of the lattice, from left to right. At time step $t = 0$ this cell is in state 0 and its neighbourhood is 101. For this situation, the state transition rule depicted at the top of the figure establishes that the new state of that cell has to be 1. The updating is made for all cells of the lattice, in a synchronous fashion. Therefore, the lattice configuration shown at the bottom of the figure is obtained at time step $t = 1$.

Lattice evolution through several time steps can be shown by a spatial-temporal diagram as in Figure 3a. However, this diagram is better visualized using filled cells to represent state 1 and empty cells to state 0, as in Figure 3b.

The 256 one-dimensional, $k = 2$, $r = 1$ CA are known as the Elementary Cellular Automata (ECA). Wolfram proposed a numbering scheme for ECA, in which the output bits are lexicographically ordered, and read right-to-left, to form a binary integer between 0 and 255 [30].

The 256 elementary CA form the Elementary Rule Space [31]. Figure 4 shows the temporal evolution of four different ECA.

It is possible to evidence that they exhibit very distinct dynamics although all of them are based on simple cellular automata rules of radius 1 like the rule presented in Figure 2.

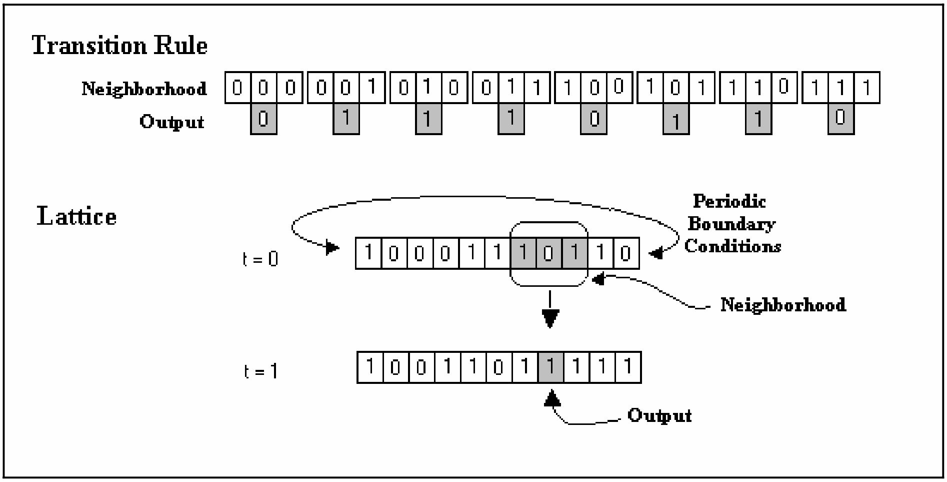
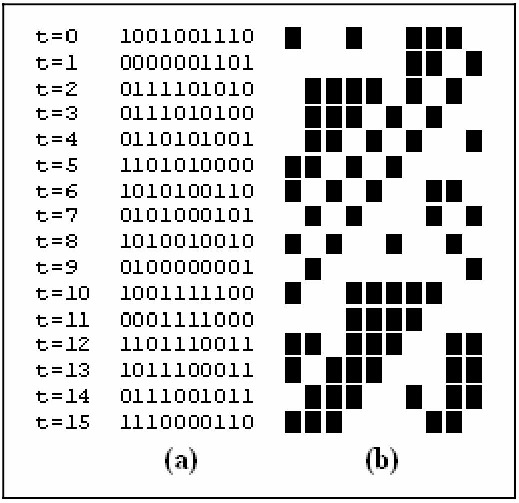


Figure 2. One-dimensional radius 1 CA



```
t=0   1001001110
t=1   0000001101
t=2   0111101010
t=3   0111010100
t=4   0110101001
t=5   1101010000
t=6   1010100110
t=7   0101000101
t=8   1010010010
t=9   0100000001
t=10  1001111100
t=11  0001111000
t=12  1101110011
t=13  1011100011
t=14  0111001011
t=15  1110000110
```

Figure 3. Equivalent spatial-temporal diagrams: (a) using states 0 and 1 (b) using filled and empty cells.

2.1. CA Dynamics and Rule Space Parameterisation

Through the analysis of the dynamic behavior exhibited by CA, it can be verified that they can be grouped into classes. A few rule space classification schemes have been used in the literature; for instance, Wolfram proposed a qualitative behavior classification, which is widely known [30]. Later on, Li and Packard proposed a series of refinements in the original Wolfram classification, one of them [31] that divides the rule space into six classes: Null, Fixed Point, Two-Cycle, Periodic, Complex (or Edge of Chaos) and Chaotic.

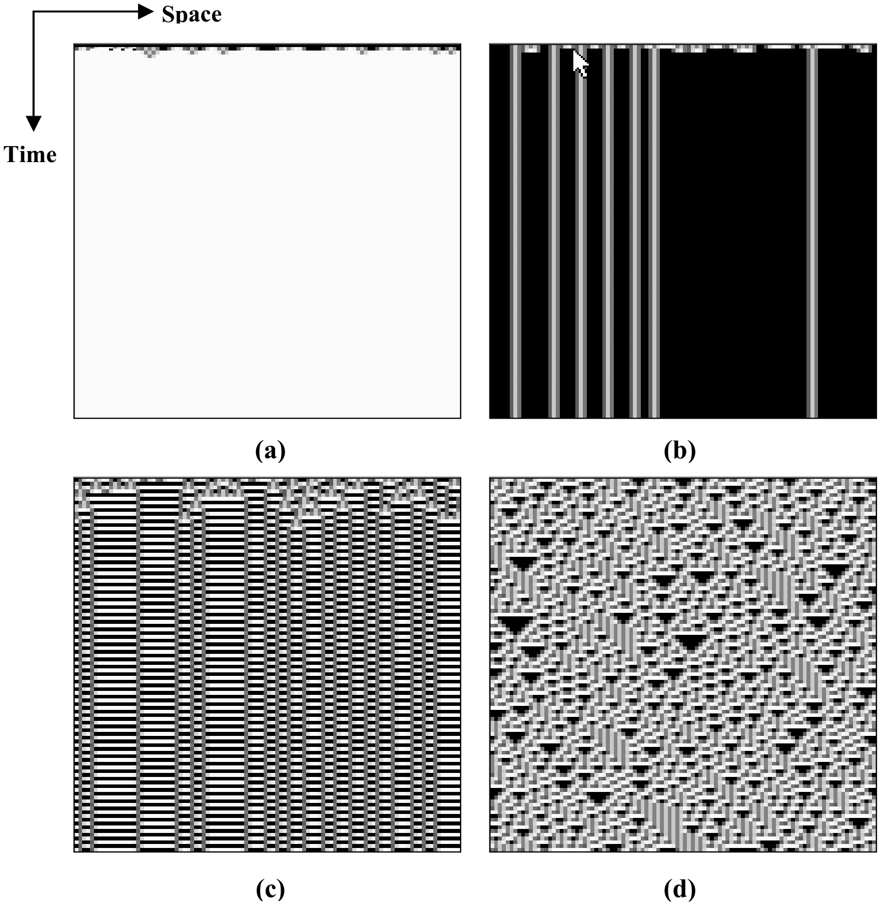


Figure 4. Different dynamic behaviours in elementary cellular automata: (a) Null, (b) Fixed Point, (c) Two-Cycle, and (d) Chaotic. In each display, times increases downwards, and space (the sequence of cells) is represented horizontally.

- **Null**. The limiting configuration (that is, the sequence of state values of all the cells in the lattice after some time steps) is only formed by 0s or 1s (Figure 4a).

- **Fixed Point**. The limiting configuration is invariant (with possibly a spatial shift) by applying the cellular automaton rule, the null configurations being excluded (Figure 4b).

- **Two-Cycle**: The limiting configuration is invariant (with possibly a spatial shift) by applying the rule twice (Figure 4c).

- **Periodic**. The limiting configuration is invariant by applying the automaton rule L times (L>2), with the cycle length L either independent or weakly dependent on the system size.

- **Edge of Chaos, or Complex**. Although their limiting dynamics may be periodic, the transition time can be extremely long and they typically increase more than linearly with the system size. This kind of behavior is difficult to observe in ECA, but a well-known example of this kind of behavior is elementary cellular automaton 110.

- **Chaotic**. These rules are characterized by the exponential divergence of its cycle length with the system size, and for the unstability with respect to perturbations (Figure 4d).

The dynamics of a cellular automaton is associated with its transition rule. In order to help forecast the dynamic behavior of CA, several parameters have been proposed, directly calculated from their transition table [10], [31], [32], [33], with more than a single parameter being usually used at a time. It should be remarked, however, that it is not possible to precisely forecast the dynamic behavior of a generic cellular automaton, from an arbitrary initial configuration [34]; all that can be expected is a parameter set that can *help* forecast its dynamic behavior.

Along this line, a set of five parameters was selected in [10], two of them drawn from among those already published, and three new ones. In the experiments involving the DCT task described in the next section, four parameters from that set were used: Sensitivity [33], Absolute Activity [10], Neighbourhood Dominance [10] and Activity Propagation [10]. All of them have been normalized between 0 and 1, for one-dimensional CA with any radius, and they are formally defined in [10].

For the rest of the section, as a matter of supplementary information, the four parameters used here are now briefly and informally described, and then a summary is provided of how they correlate to cellular automata dynamical behavior. The level of information provided below is sufficient for present purposes; the formal definition of the parameters and much more detailed information about them can be found in reference [10]:

- **Neighbourhood Dominance** quantifies how much change is entailed by the CA rule transitions, in the state of the centre cell, in respect to the state that predominates in the neighbourhood as a whole (for example, in the state transition 010 → 0, neighbourhood dominance occurs because the state that predominates in the neighbourhood is “0” and the transition maps the centre cell state onto “0”). Accordingly, the parameter value comes from a count (in fact, a weighed sum) of the number of transitions of the CA rule in which neighbourhood dominance occurs, with the additional feature that, the more homogeneous the neighbourhood involved, the higher its weight.

- **Absolute Activity** quantifies how much change is entailed by the CA rule transitions, in the state of the centre cell, in relation to two aspects: the state of the centre cell of the neighbourhood, and the states of the pair of cells which are equally apart from the centre cell (that is, the pair of nearest neighbours of the centre cell, then the pair of next-nearest neighbours, and so on).

- **Sensitivity** quantifies the effect, in the output bit of the state transitions, of flipping a single bit in the neighbourhood. It is defined as the number of flips in the output bit of the cellular automaton rule, caused by flipping the state of a cell of the neighborhood, each cell at a time, over all possible neighborhoods of the rule.

- **Activity Propagation** provides a way to quantify, at the same time, both the *neighbourhood dominance* and the *sensitivity* of each CA rule transition. It was defined from two concepts related to the definitions of these previous cited parameters: the possibility of a transition "following" (or not) the state that dominates the neighborhood, and the possibility of a transition being sensitive to a minimal change (a single state flip) in the neighborhood. More specifically, the parameter quantifies if the neighborhood dominance is sensitive to changing the state of just one cell of the neighborhood. For example, in the state transition 010 → 0 neighborhood dominance occurs, but it is not sensitive to a change in the leftmost cell, if the rule also has the transition 110 → 1; in other words, for this rule, even flipping a bit of the original neighborhood, the neighborhood dominance remains.

Interesting results were obtained in the characterization of the elementary space with this set of parameters [10]. Figure 5 presents four charts where the relative occurrences of rules with selected dynamical behaviors (null, chaotic, fixed point and two-cycle) in the elementary space are plotted.

The main features that led to the selection of each parameter are summarized next: *sensitivity* helps to relatively discriminate null and chaotic behaviors (Figure 5a); *absolute activity* and *neighbourhood dominance* help in the relative discrimination between fixed point and two-cycle behaviors (Figures 5b and 5c); and *activity propagation* helps define the region characterized by null and fixed point rules, grouped as *fixed* rules (Figure 5d). Figure 5 refers only to elementary cellular automata: one-dimensional binary CA with radius 1.

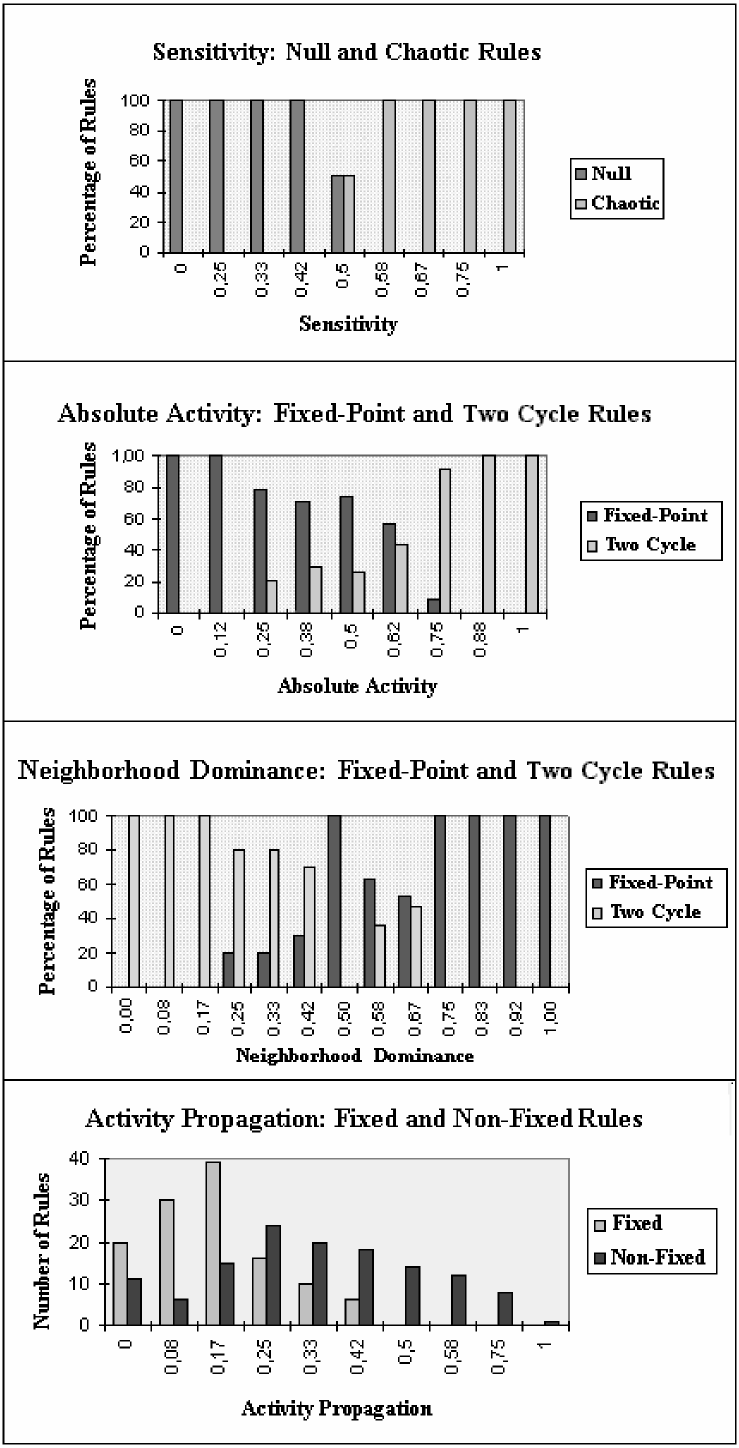


Figure 5. Relative occurrences of the elementary CA rules to each value of the parameter: a) Sensitivity b) Absolute Activity c) Neighbourhood Dominance d) Activity Propagation.

2.2. Computational Tasks and Evolution of Cellular Automata

Cellular automata can model complex systems, as well as execute computations in a non-standard fashion. Various investigations have been carried out on the computational power of CA, with concentrated efforts in the study of one-dimensional CA capable of performing computational tasks [1].

The most widely studied CA task is the Density Classification Task (DCT) [3], [4], [5], [6], [7], [35], [36]. In this task the objective is to find a binary one-dimensional cellular automaton that can classify the density of 1s in the initial configuration of its lattice, such that: if the initial lattice has more 1s than 0s, the automaton should converge to a null configuration of 1s, after a transient period; otherwise, it should converge to a null configuration of 0s.

Figure 6 shows two examples of evolution of a cellular automaton performing the DCT. This rule was found in the experiments to be described in next sections. While in the first example about 48% of the cells in the initial configuration are 1s, in the other example there are more 1s than 0s (about 53%) at step 0. Notice that, in terms of dynamical behavior, DCT rules are of the Null type.

Once a computational task is defined, manual programming is difficult and costly, and exhaustive search of the rule space becomes impossible, due to its high cardinality. A solution is the use of search and optimization methods, particularly evolutionary computation methods [3], [4], [5], [6], [7], [35], [36].

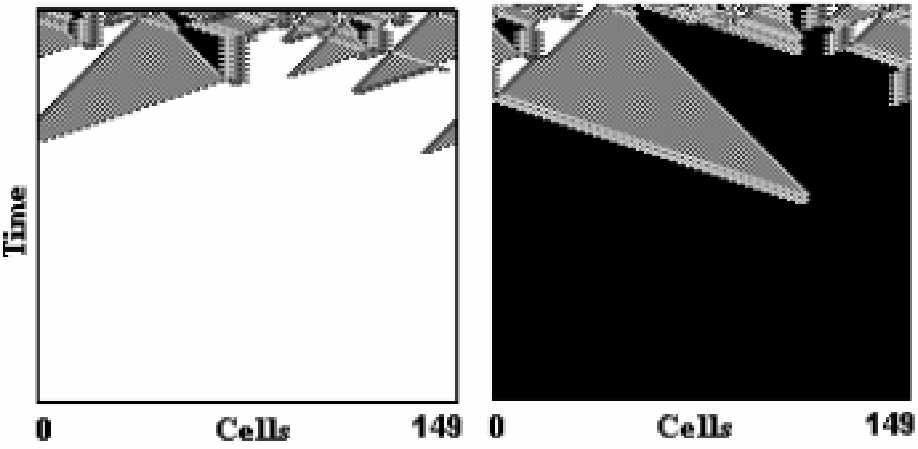


Figure 6. Density Classification Task performed by rule 070447470700460705774757F777FF77.

Packard (1988) was the first to publish results using a genetic algorithm as a tool to find CA rules for the DCT [35]. He considered one-dimensional CA rules as individuals in a population and defined their fitness according to their ability to perform the task. Crossover among two CA was defined by the creation of two new transition rules, out of segments of two other rules; mutation was achieved by randomly flipping the output bit of one of the transitions of the rule. Other evolutionary computation techniques were used to find such kind of CA rules [4], [5], [6], [8], [36]. This evolutionary approach was also applied to the two-dimensional version of DCT [37], [38].

2.3. Previous experiments employing a parameter-based heuristic in DCT search

This section summarizes some of related works that involves the use of the forecast parameters presented in section 3 to guide the genetic search for DCT rules. Some of them were realized in a simple GA environment and others in a multiobjective GA environment. An important work about evolution of CA rules to perform DCT was present in [3]. The original framework proposed by Mitchell and collaborators was adopted in several subsequent works [4], [5], [6], [7], [8], [36].

In a previous work, we first implemented a simple GA environment based on the framework proposed by Mitchell and colleagues [3]. Subsequently, we modified the environment so as to incorporate a heuristic based on the selected forecast parameters cited in the previous section [7]. The parameter-guided GA was then used to find DCT rules. First, parameter value regions where good rules should be more likely to occur were obtained by calculating the parameter values for some published CA rules, the following ranges having been obtained:

- Sensitivity: 0.23 to 0.40.
- Neighbourhood Dominance: 0.84 to 0.91.
- Absolute Activity: 0.10 to 0.26.
- Activity Propagation: 0.07 to 0.11.

This was the information that was used as an auxiliary metric to guide the processes underlying the GA search, as described below.

A simple GA was adapted in [7] so as to incorporate the parameter-based heuristic in two aspects. First, the fitness function F of a cellular automaton rule was defined as a weighed composition between the heuristic-based component (Hp) and the fitness component derived from the actual performance of the rule (F_{IC}) in the attempt to solve the DCT; Equation 1 clarifies this issue.

$$F = F_{IC} + \rho \times Hp \tag{1}$$

The parameter-based heuristic is coded as a function that returns a value between 0 and 100 for each cellular automaton rule, depending on the values of the rule parameters. More precisely, the function returns 100 if all parameter values of the cellular automaton rule match those of the ranges of the published rules; otherwise, the value returned decreases linearly as the parameter values deviate from those ranges (see details in [7] and [10]).

In the work reported in [7], all parameters contribute equally to the calculation of the heuristic component fitness. Changing this approach is precisely one of the points being made in this paper (as will be clear in the next section, in the multiobjective experiments with Decomposed Heuristic).

The function F_{IC} also returns a value between 0 and 100, according to the rule efficacy in solving 100 different initial configurations, randomly generated with uniform distribution of 1-bit densities [3]. Finally – and crucially, to the first point being made in this paper – ρ is the weight that establishes the influence of the heuristic component in the overall rule fitness.

The second aspect in which the heuristic information was used in the genetic search was in that it allowed the definition of biased genetic operators of reproduction and mutation. Basically, at the time of selecting the crossover point and the rule table bits to be mutated, N_{CM} crossover points are chosen and N_{CM} mutations are made. Among the individuals generated – 2 N_{CM} offspring out of the crossovers, and N_{CM} out of the mutations – the selected offspring pair is the one representing rules with the highest individual *Hp* values; analogously, the selected mutated rule is also the one leading to the highest *Hp* among the N_{CM} possibilities.

In the experiments reported in [7], with ρ = 40% and the N_{CM} value above, and the insertion of the parameter information managed to improve the performance of the rules found for the density classification task, both in average and in respect to the best rules found. The parameter-based approach was also used in the evolution of CA rules for the two-dimensional version of DCT return good results [38].

Since an analysis of the effect of the weight ρ in the context of the DCT has not been done in [7], a series of experiments were performed with varying values of ρ from 0% up to 120%. The results of these experiments are detailed in [39]. It was possible to observe that all experiments with the parameter-based heuristic yielded superior results in comparison with the experiment without the heuristic. However, an adequate value of ρ is required to improve the search, especially when considering the high efficacy rules obtained within each experiment. Within the values tested, the best experiment relied on ρ = 40%, as used in [7]. Nevertheless, a question remains of whether there may be another ρ value which might yield better results and whether that value would be adequate for other parameter ranges and other computational tasks.

In a second stage of the work, instead of using the weighed sum of Equation 1, a multiobjective approach was used; in this case, the parameter-based heuristic is still introduced in crossover and mutation in the way described at the end of the last section (which is the same way as in [7]). However, the rule fitness is no longer given by the composition of the two separated objectives F_{IC} and *Hp* from Equation 1: instead of establishing any *a priori* heuristic weight, the multiobjective dynamics are expected to define the relative importance of the two objectives during the search [39].

A multiobjective (MO) environment was then implemented after the NSGA method [18], so that instead of the composite fitness given by Equation 1, the non-domination classification and sharing fitness of the MO environment were used for the fitness evaluation of a candidate solution. The other steps are the same as those used in the original experiment [3], namely, elitism of 20%, with the crossover pairs being randomly selected directly from the elite. The two objectives, F_{IC} and *Hp* were implemented in the same way as described earlier [7].

At this point, it is important to emphasize the nature of the two objectives, F_{IC} and *Hp.* Note that both can be seen as estimates of the GA run final efficacy. After all, while F_{IC} is measured in the smaller sample of 200 ICs, generated with uniform distribution (differently from the binomial distribution of the final evaluation), *Hp* is an heuristic that can only be thought of as hinting at a region in the parameters space where good rules are more likely to occur. Note also that the objectives alone cannot guarantee that a rule with a high value along them is better than another rule with a lower value; but, here, unlike a typical MO problem and regardless of the number of objectives involved, there is an evaluation that can be used to define the real best solution, that is exactly the efficacy in a sample of 10^4 ICs randomly generated. Note that this evaluation is not used during the evolution as an objective; it is applied only to test the quality of the final best rule found in the final population.

The detailed results of the multiobjective experiments were represented in [39]. Three experiments were performed; each one obtained out of 200 runs. The following GA parameters were used: 200 individuals in the populations, 200 initial configurations (uniform distribution) for testing the rules at each generation, and 1000 generations per run. One experiment was performed with the simple GA without the parameter-based heuristic, named WH_2 experiment. In the second experiment, a simple GA was also used but with the parameter-based heuristic incorporated as in Equation 1 with ρ = 40%; this was called ρ40_2 experiment. Finally, the experiment MO_2 was performed with the NSGA-based environment with the heuristic information.

We will reproduce here some results of there three experiments reported in reference [39] only to facilitate the comparison with the new experiments reported in the next section. All experiments were composed by 200 GA runs, and the efficacies of the 10 best rules found in each experiment are presented in Figure 7, extracted from [39]. It was possible to observe that the multiobjective experiment with the composed heuristic (MO_2) outperformed the other two that uses a single GA. This superiority was also confirmed by the average of the efficacies found in 200 runs: 78.73 (WH_2), 80.54 (ρ40_2) and 81.16 (MO_2).

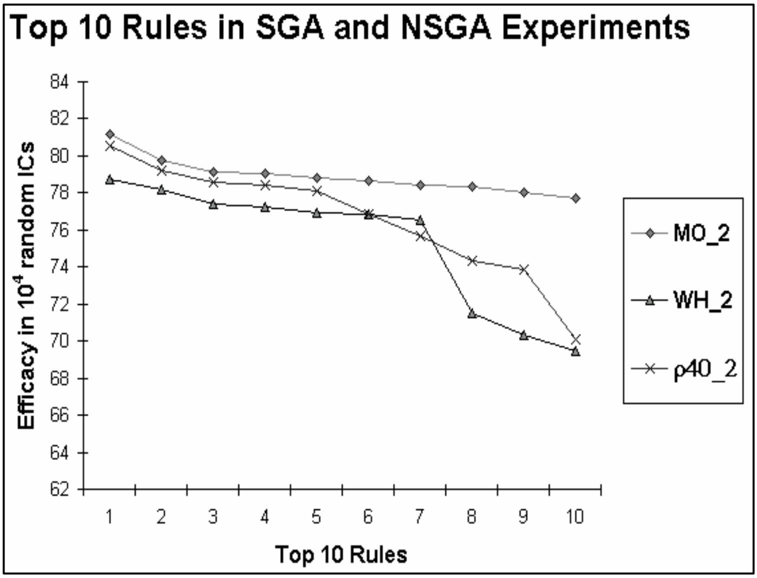


Figure 7. Efficacy of the top 10 rules with NSGA and decomposed heuristic, with 200 individuals, 1000 generations and 100 runs (extracted from [39]).

3. Experiments with Decomposed Parameter-Based Heuristic

This section presents the results of several experiments involving the evolution of CA rules to perform DCT. All the experiments reported here are based on the original framework proposed by Mitchell and collaborators (1993) [3] and adopted in several subsequent works [4], [5], [6], [7], [8], [36].

In order to clarify the approach, the major points are explained here: the population is formed by 100 state transition rules of binary CA with radius 3, which are represented by 128-bit strings. The new population at each generation is created with the 20% elite of the previous generation, the other 80% being formed by new 128-bit strings generated through single-point crossover over the selected pairs. The mating pairs are randomly chosen from the elite. All the new individuals are submitted to mutation, which consists of complementing some bits of the rule at a probability rate of 2%. The fitness evaluation in the single objective approach – named F_{IC} – is made by counting the success of an individual (a candidate rule) in classifying a sample of 100 random initial lattices formed by 149 bits, uniformly distributed. Every rule is applied over each initial lattice, and after 300 time steps the lattice is checked in terms of whether it converged to the desired configuration: all 0-bits or 1-bits, depending on the initial density. Therefore, the fitness evaluation returns an integer between 0 and 100, representing the percentage of correct classifications. A better convergence was observed by Mitchell and colleagues [3] when both the 128-bit strings of the initial population and the 149-bit strings of the 100 initial lattices sampled at each generation are formed according to a uniform distribution (that is, all 1-bit densities evenly spread out). The final efficacy of each run was measured by testing the performance of the best rule found, at the end of the run, in the classification of 10^4 random initial configurations, sampled according to a binomial distribution (that is, each bit is independently chosen with probability 0.5). This causes the configurations having 50% 1-bits to be the most represented one in the sample, a situation which happens to be the most difficult case in the task.

In the multiobjective experiments reported in [39] and summarized in the previous section, although the heuristic information was considered as an independent objective, it is in fact composed of four different parameters, each of them contributing equally for the fitness component H_P (through a simple average of the four individual contributions). However, we believed that the individual guidance role of each parameter might have distinct effectiveness over the genetic search.

In our previous works with the parameter-based heuristic, the contribution of each parameter had not been evaluated, the major barrier having been that, in order to go about it with the simple objective approach, the discovery of a good weight for each one of the parameters is required. However, the composed heuristic allowed the parameter-based heuristic to be broken into different and independent objectives, associated with each individual parameter. This could be done without the need to establish a weight to each objective, thus allowing the evolutionary multiobjective algorithm to directly handle the resulting multi-directed search. Some experiments in which the parameter-based heuristic was *decomposed* are described below.

Fifteen multiobjective experiments were performed, where one of the objectives is always the actual performance of the rule (F_{IC}) in the attempt to solve the DCT. In addition to F_{IC}, one or more objectives are then used, depending on the number of parameters considered. With this rationale, four categories of experiments were performed, as follows:

- Two objectives: F_{IC} and the value returned by the heuristic, considering each one of the four parameters individually – Sensitivity, Neighbourhood Dominance, Absolute Activity, and Activity Propagation – thus yielding four experiments.

- Three objectives: F_{IC} and the values returned by two separate heuristics, each one considering a single individual parameter. Since the four parameters were combined two by two, six experiments of this type were carried out.

- Four objectives: F_{IC} and the values returned by three separate heuristics, each one considering a single individual parameter. And since the four parameters were combined three by three, four experiments become possible.

- Five objectives: F_{IC} and the values returned by four separate heuristics, each one considering each individual parameter, thus yielding only one experiment.

The fifteen experiments were carried out using the specifications presented in Table 1.

All of them were performed in the multiobjective (MO) environment implemented after the NSGA method [18], in which the non-domination classification and sharing fitness of the MO environment were used for the fitness evaluation of a candidate solution.

The experiments of each category that yielded the best results are:

- Two objectives: F_{IC} + Sensitivity, named MO_S.

- Three objectives: F_{IC} + Sensitivity + Activity Propagation, named MO_SP.

- Four objectives: F_{IC} + Sensitivity + Activity Propagation + Neighbourhood Dominance, named MO_SPN.

- Five objectives: F_{IC} + Sensitivity + Neighbourhood Dominance + Absolute Activity + Activity Propagation, named MO_SPNA.

The results of the four multiobjective experiments with the decomposed heuristic are presented in Table 2 and Figure 8. The table also presents the results of the MO experiment, which used two objectives: F_{IC} and the composed heuristic H_P, obtained through a simple average of the four parameters. MO experiment used the same specification in Table 1 and it was previously reported in [39].

The efficacy of each run was measured by testing the performance of the best rule found, at the end of the run, in the classification of 10^4 random initial configurations. Columns 2 through 7 in Table 2 display the percentage of runs in which the efficacy of the best rule found was within the corresponding interval shown therein. For example, the second column shows the percentage of runs (in a total of 200) in which the efficacy of the best rule found was within the interval from 45% to 55%.

Table 1: Parameter values used in the initial experiments.

Number of individuals	100
Number of generations	100
Number of ICs per generation	100 (uniform distribution)
Number of ICs at the final evaluation of the run	10^4 (binomial distribution)
Number of GA runs	200
Elitism rate	20%
Crossover selection	Random selection out of the elite
Mutation rate	2% per bit
CA radius	3
Number of CA cells	149
Number of CA steps	300
Fitness sharing radius (NSGA)	0.5

Table 2. Efficacy ranges and other results obtained with NSGA with composed heuristic (MO) and decomposed heuristic (MO_S, MO_SP, MO_SPN and MO_SPNA), with 100 individuals. For every experiment, the table gives: percentage of runs in which the efficacy of the best rule found in the experiment was within the corresponding interval shown; efficacy of the best rule found; the average (μ) efficacy and the corresponding standard deviation (σ), considering all runs; and again the latter, but only for the 10 best rules in each experiment (the Top10 rules).

	[45,55)	**[55,60)**	**[60,65)**	**[65,70)**	**[70,75)**	**[75,80)**	**Best Rule**	**μ (all)**	**σ (all)**	**μ (T10)**	**σ (T10)**
MO	0.0	0.4	17.8	79.2	1.0	1.6	78.9	66.3	2.1	76.6	1.4
MO_S	26.0	0.2	22.8	45.8	3.4	1.8	79.1	62.0	7.4	77.1	1.2
MO_SP	0.0	0.2	24.0	68.4	3.8	3.6	78.8	66.4	2.8	77.2	0.9
MO_SPN	0.0	0.0	21.8	73.8	2.2	2.2	78.0	66.4	2.3	76.9	0.9
MO_SPNA	0.0	0.2	21.2	75.0	2.0	1.6	77.8	66.2	2.1	76.0	0.9

Additionally, Table 2 shows the best rule found, the average efficacy and the corresponding standard deviation considering the total of runs, and also considering only the 10 best rules in each experiment (the Top10 rules).

For present purposes the best criterion for comparing the quality of the results across the various search schemes is their ability to find rules with better efficacy, which can happen in the form of the best rule found – or the top rules found, say, the top ten rules – throughout the runs, and in the ability to find more rules in the highest efficacy ranges. However, it is not clear how these distinct quality variables should be combined so as to yield a single variable. If we relax on that, sticking to a more qualitative evaluation based on the latter three quality variables, comparison becomes easier, while, we believe, still truthful.

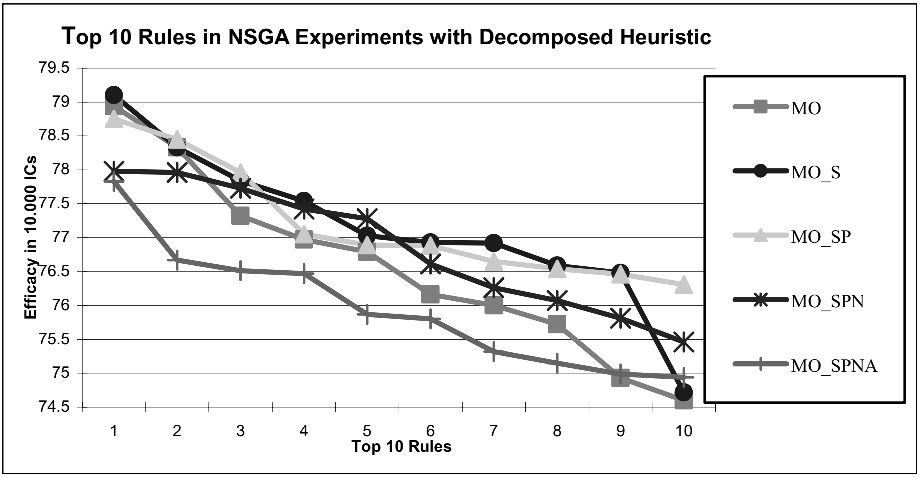


Figure 8. Efficacy of the top 10 rules with NSGA and decomposed heuristic, with population of 100 individuals.

Accordingly, one can observe that the experiment using the four decomposed parameters (MO_SPNA) is worse than the one that uses the four parameters in a composed heuristic (MO). Besides, MO is equivalent to MO_SPN and a little worse than MO_SP. On its part, MO_S yields good results, although it returns the worst results when considering the average of all the runs. And finally, MO_SP is the experiment that yields the best results out of all those with population size of 100 individuals.

Subsequently, further experiments were performed with the decomposed heuristic; so that the search could run freer, the idea being to check whether the NSGA-based environment could better explore the search space in less strict conditions. The following parameters were used: 200 individuals in the populations, 200 initial configurations (uniform distribution) for testing the rules at each generation, and 1000 generations per run. We discarded only the experiment MO_SPNA at this phase.

The new experiments were named: MO_S_2, MO_SP_2 and MO_SPN_2. All of them were composed of 100 GA runs, and the results are presented in Table 3 and Figure 9. MO_2 experiment was performed with the composed heuristic and the same specifications above; its results were extracted from [39]. Since it had been published with 200 runs in [39] (as the results reproduced in Figure 7) in order to allow its comparison with the results with the new decomposed heuristic experiments (now using only the first 100 runs), Table 3 and Figure 9 show the MO_2 results under the new name of MO_2*.

One can see that the results with the decomposed heuristic experiments using the more flexible parameter settings entailed a noticeable improvement in relation to the one with the composed heuristic (MO_2*). As a matter of fact, the decomposed heuristic experiments entailed the best performance as a whole, even when comparing 200 runs of the composed heuristic experiments with population size 200 (Figure 7) [39], with the results of MO_S_2, MO_SP_2 and MO_SPN_2 with only 100 runs. Figure 9 displays the Top10 rules for all these experiments.

All in all, the following summary applies:

- All experiments with the parameter-based heuristic, multiobjective or not, have performed better than the plain GA search WH_2 (Figure 7), reported in [39].

- The MO experiments, composed or not, yielded better results than the GA-based experiment with the best compounded fitness *ρ40*_2 (Figure 7), reported in [39].

- Three MO experiments with decomposed heuristic (MO_S_2, MO_SP_2 and MO_SPN_2) entailed a significant improvement over the MO experiment with composed heuristic MO_2 (Figure 7), reported in [39].

It is very simple to explain why the heuristic-based approach outperforms the pure GA: the information contained in the parameter values is really relevant for indicating promising regions of the search space, so that the search processes can usefully exploit them, regardless of the degree of sophistication in which they are accounted for.

It is also simple to explain why multiobjective approaches should perform better than the weighed version of the GA: the multiobjective approach avoids the arbitrariness of the ρ-weight between F_{IC} and Hp that defines the joint fitness function. And it is easy to accept why the decomposed heuristic can be advantageous over the composed version: by decomposing the heuristics more room is left for evolution to exploit the individual indications of each parameter (this is in fact the main hypothesis underlying the present work).

Table 3. Efficacy ranges and other results obtained with the simple GA and NSGA, with 200 individuals, 1000 generations and 100 runs.

	[60,65)	[65,70)	[70,75)	[75,80)	[80,85)	Best Rule	μ (all)	σ (all)	μ (T10)	σ (T10)
MO_2*	2.0	85.0	6.0	7.0	0.0	79.2	68.6	3.0	76.1	3.4
MO_S_2	5.0	57.0	6.0	16.0	16.0	82.7	71.5	5.8	81.4	0.7
MO_SP_2	2.0	51.0	8.0	28.0	11.0	82.7	72.4	5.7	81.0	0.8
MO_SPN_2	2.0	61.0	6.0	25.0	6.0	81.7	71.5	5.1	80.5	1.0

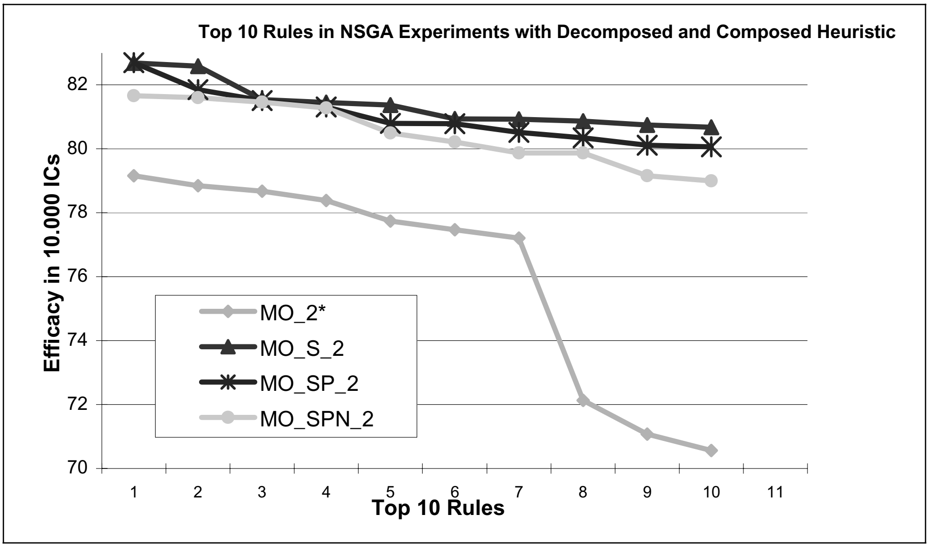


Figure 9. Efficacy of the top 10 rules with NSGA and decomposed heuristic, with 200 individuals, 1000 generations and 100 runs.

What is hard to explain is why the use of a certain single parameter, or of a specific parameter combination, turns out to be more effective than the use of an alternative single parameter or combination. Many issues can be involved in this: the specific ways the parameter or parameter combination act in a particular search space, the specific features of the search space, details of the algorithm at use, the possible CA strategies for solving the problem at issue, etc. Although we have pursued explanations of this sort, no general statement can be made to this respect so far.

4. Computational Strategies

In the previous section, only the efficacy of the rules found was discussed, as measured by the percentage of ICs (in a sample of 10^4, randomly generated through a binomial distribution) which the CA rules could correctly classify. However, an important characteristic not discussed so far is the type of strategies used by the CA in their attempt to classify an IC. Mitchell and collaborators (1993) identified and discussed three main strategies found in their experiments with the DCT [3]:

- **Default**: Simple null rules emerge, that classify all initial configurations in a specific class (class 0 or class 1), regardless of the initial density. Since approximately 50% of the sample may in fact be mapped onto the specific class, this strategy leads to CA rules with about 50% efficacy.

- **Block-expanding**: Every initial configuration is classified in a specific class (class 0 or class 1), independently of its density, unless the IC contains a sufficiently large block of cells in the complementary class; typically, this means blocks of at least the size of the neighborhood at issue, that is, 7 cells. Such a strategy is slightly more refined than the default one, usually leading to rules with efficacy between 55% and 72% (for one-dimensional, radius 3 CA, with 149 cells in the lattice).

- **Particle**: The initial configurations are classified according to the result of interactions between meso-scale (particle-like) patterns that can appear along the CA temporal evolution. This strategy typically leads to CA rules with efficacy above 75% (for the kind of CA mentioned above).

The three kinds of strategies may occur in any GA run. However, in the experiments reported in [3] performed with a standard genetic algorithm, approximately 10% of the runs could find only default strategies, while 87% of the runs led to block-expanding strategies, after default strategies had been found in the initial generations. As for the remaining 3% of the runs, in these cases the populations could progressively evolve from default strategies to block-expanding strategies, until discovering rules with particle strategies.

In tune with the latter, the quantitative, efficacy-oriented analyses of our experiments carried out in the previous sections are now extended through a discussion in respect to the computational strategies performed by the best rules found.

After examining all CA rules with efficacy above 70% found in the experiments MO_S_2, MO_SP_2, MO_SPN_2, MO_2*, ρ40_2* and WH_2* – remember that the *-marker in an experiment denomination means that only its 100 first runs are being taken into account from the experiments reported in [39] – we observed the spatial-patterns generated for each rule over a sample of ICs, and then identified the kind of strategy employed by the rule. In respect to the particle strategy, the most complex one in computational terms, each experiment could find a different number of rules, 123 in total, as follows:

- WH_2*: 3 rules
- ρ40_2*: 7 rules
- MO_2*: 7 rules
- MO_S_2: 32 rules
- MO_SP_2: 40 rules
- MO_SPN_2: 32 rules

The three best rules with particle strategy at each experiment are listed in Table 4, together with their hexadecimal code, their efficacy in a sample of 10^4 random ICs, and the experiment in which they have been found.

Figure 6 shows two spatial-patterns generated by applying the best particle-based rule found (070447470700460705774757F777FF77), considering all the experiments discussed, which was found in the MO_SP_2 experiment. As it is now clear, the multiobjective experiments using decomposed parameter-based heuristic are not only able to find CA with better efficacy (as the previous section has shown), but they also yield more examples of CA that employ the particle strategy.

Figure 10 plots, for each of the 100 runs, the efficacy of the best rule found in that run for each experiment (similarly to the representation used in [36]). The values are ordered according to decreasing efficacy values. The symbols denote which strategy used by the CA: particle (▲) or block-expanding (●).

Table 4. Three best particle CA found in each experiment.

Rule	E (%)	Experiment
015E0012005500571F5FFFFF0F55CF5F	82.7	MO_S_2
10111000531512531F15FF5FDF5DDF5F	82.6	
00010355015511571F150F77FFF5FF57	81.5	
070447470700460705774757F777FF77	82.7	MO_SP_2
015400550050045F055FFFDF5557FF5F	81.9	
0445004C37770E3F044500CDF7773FFF	81.5	
1500500035007707155377 5FF5F77F7F	81.7	MO_SPN_2
000104171DDF555704DF441FDDDFD557	81.6	
0100003001131137 0DFFBFFBDDFF11FF	81.5	
0001090703030B031F1F6F37FF776F77	79.2	MO_2*
0100050D1D9D155F05FD555FDDFF5557	78.9	
00010302111011317F5FFFFDDFF11FF	78.7	
0001017D2113C35F4B15DF75275B9FD7	78.8	ρ40_2*
015500400054563F1057BF0FB7FFFB7F	77.8	
0071023C00224D170379B53747BFFF7F	77.3	
10041383005313DD3357CFED875F1FDF	78.7	WO_2*
040305502F06457D05013757D5F7FF7F	78.2	
05047000006516D053FF5FF977FE77F	77.3	

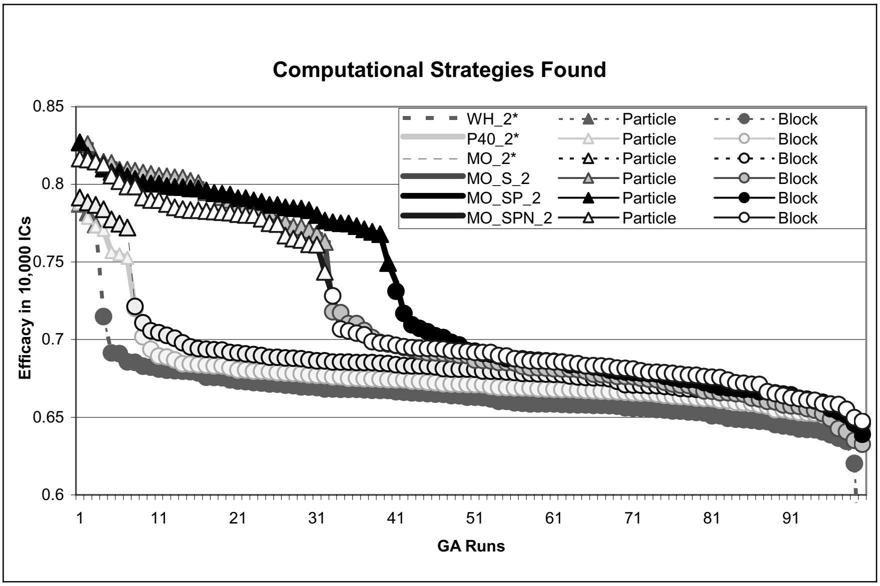


Figure 10. Efficacy and strategy of the best rule found in each run for all experiments with 200 individuals.

5. Final Remarks

The experiments have shown that the multiobjective solution is a good approach for the incorporation of the parameter-based heuristic we have used in previous works, as a way to help automatic CA programming. The results obtained with the multiobjective approach are, clearly, at least as good as the one derived from the adequate choice of a weight to balance the role of the heuristic, with the advantage that no actual choice has to be made.

The lesson the experiments discussed here bring forth is that the notion of Pareto dominance seems to be prevailing also in the context of the parameter-based heuristic in CA search.

Besides avoiding the definition of an a priori relative weight of the heuristic in the search, the multiobjective approach also allowed us the opportunity to experiment with several options for the formation itself of the heuristic, differently from its original conception [7], where the parameters contributed equally to the search. Such a kind of investigation in the traditional approach (with the weighted average) would bring a much larger level of complexity to the problem, since up to five weights would have to be defined.

By comparing the last three experiments, one can notice that MO_S_2 (with 2 objectives) and MO_SP_2 (3 objectives) outperformed the one with 4 (MO_SPN_2). A remaining question, though, is whether these results are due to more relevant information embedded in the Sensitivity parameter when compared to the others, or whether it is due to shortcomings of the multiobjective environment, insofar as a larger number of parameters have to be handled.

While, at least in theory, EMOO methods are able to account for any number of objectives, the practical difficulty observed herein for handling more than 2 objectives has been previously pointed at by other researchers [40]. More recent methods, such as SPEA [22] and SPEA2 [23] have been proposed with the motivation for also coping with such a kind of deficiency.

In tune with that, as a follow-up investigation, we are in the process of adapting the CA parameterised search scheme, so that more recently proposed multiobjective algorithms can also be tried out. Hopefully our results may continue improving, as they did with the NSGA-based multiobjective decomposition of heuristics discussed in the paper, when compared with the other approaches we tried previously.

We consider the results presented in sections 3 and 4 a substantial evidence of the efficacy of the approach proposed in this work. However, since genetic algorithms are stochastic methods it is better to provide the results with statistical confidence. In fact, empirically evaluating the accuracy of hypotheses is fundamental to machine learning. A statistical method for estimating hypotheses accuracy can be used in the experiments related here to show that proposed methods really have better results than previous used techniques. In a further work, we intend to evaluate the hypotheses accuracy of experiments of such kind. For more details in evaluating hypotheses in machine learning refers to reference [40].

The quest for the best possible (imperfect) rule for the DCT thus remains, as a challenging pursuit. And since we had to wait for nearly 20 years until the DCT was proven not to be solvable, waiting for an analytic solution to the question is not appealing; a more fruitful alternative seems to be the empirical way of carrying on developing progressively more sophisticated algorithms and heuristic design methods, and setting them on trial. The possibilities are that we would then be able to transfer knowledge acquired in the DCT to similar inverse problems, opening the possibility of designing CA rules able to perform other computational tasks, thus availing ourselves with valuable data for increasing our understanding of how cellular automata are able to compute.

Acknowledgements

This work was made possible primarily due to research grants from *Fundo Mackenzie de Pesquisa* (MACKPESQUISA), particularly through Edital 2007, to whom we all are grateful. G.M.B.O. thanks *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq) for the research grant PQ 309711/2007-0 and *Fundação de Amparo à Pesquisa do Estado de Minas Gerais* (FAPEMIG) for the research grant PPM CEX-APQ 4798-5.01/07. P.P.B.O. thanks *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP) for the research grant Proc. 2005/04696-3, and Wolfram Research for the Mathematica Academic Grant No. 1149.

References

[1] Mitchell, M. (1996). "Computation cellular automata: a selected review". In Gramss, T., editor, *Nonstandard Computation*, VCH Verlagsgesellschaft, Weinheim, Germany.

[2] Wolfram, S. (2002). *A New Kind of Science*, Wolfram Media.

[3] Mitchell, M., Hraber, P. T. and Crutchfield, J. P. (1993). "Evolving cellular automata to perform computations: mechanism and impediments". *Physica D*, 75:361-391.

[4] Andre, D.; Bennett, F. and Koza, J. (1996). "Evolution of Intricate Long-Distance Communication Signals in Cellular Automata Using Programming". In Langton, C. G. and Shimohara, T., editors, *Proceedings of Artificial Life V*. Japan, MIT Press / Bradford Books, pages 16-18.

[5] Juillé, H. and Pollack, J. B. (1998). "Coevolving the "ideal" trainer: application to the discovery of cellular automata rules". In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R. L. editors, *Proceedings of Genetic Programming Conference*, pages 22-25, San Mateo, California, University of Wisconsin at Madison.

[6] Werfel, J., Mitchell, M. and Crutchfield, J. P. (2000). "Resource sharing and coevolution in evolving cellular automata". *IEEE Transactions on Evolutionary Computation*, 4(4):388-393.

[7] Oliveira, G. M. B., de Oliveira, P. P. B. and Omar, N. (2000). "Evolving solutions of the density classification task in 1D cellular automata, guided by parameters that estimate their dynamic behavior". In Bedau, M. A., McCaskill, J. S., Packard, N.H. and Rasmussen, S., editors, *Artificial Life VII*, pages 428-436, MIT Press, Boston, Massachusetts.

[8] de Oliveira, P.P.B., Bortot J.C., and Oliveira, G.M.B. (2006). "The best currently known class of dynamically equivalent cellular automata rules for density classification". *Neurocomputing*, 70(1-3):35-43.

[9] Goldberg, D. E. (1989). *Genetic algorithm in search, optimization and machine learning*. Addison-Wesley.

[10] Oliveira, G. M. B., de Oliveira, P. P. B. and Omar, N. (2001). "Definition and applications of a five-parameter characterization of one-dimensional cellular automata rule space", *Artificial Life Journal*, 7(3):277-301.

[11] Land, M. and Belew, R. (1995). "No perfect two-state cellular automata for density classification exists". *Physical Review Letters*, 74(25):5148-5150.

[12] Capcarrère, M. and Sipper, M. "Necessary conditions for density classification by cellular automata". *Physical Review E*, 64(3):6113-6117, 2001.

[13] Sipper, M., Capcarrère, M. and Ronald, E. (1998). "A simple cellular automaton that solves the density and ordering problems". *International Journal of Modern Physics*, 9(7):899-902.

[14] Fukś, H. (1997). "Solution of the density classification problem with two cellular automata rules". *Physics Review E*, 55:2081R-2084R.

[15] Capcarrère, M., Sipper, M. and Tomassini, M. "Two-state, r=1, cellular automata that classifies density". *Physical Review Letters*, 77(24):4969-4971, 1996.

[16] Collette, Y. and Siarry, P. (2003). *Multiobjective optimization: Principles and case studies*, Springer.

[17] Shaffer, J. D. (1985). "Multiple objective optimization with vector evaluated genetic algorithms". In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetics* Algorithms, Lawrence Erlbaum Associates: Mahwah, NJ, USA, pages 93-100.

[18] Srinivas, N. and Deb, K. (1994). "Multiojective Optimization using nondominated sorting in Genetic Algorithms". *Evolutionary Computation*, 2(3):221-248.

[19] Fonseca, C. M. and Fleming, P. J. (1993). "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization". In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416-423, San Mateo, California.

[20] Horn, J. and Nafpliotis, N. (1993). "Multiobjective optimization using the Niched Pareto Genetic Algorithm". *IlliGA1 Technical Report 93005*, University of Illinois at Urbana-Champaign, Urbana, Illinois.

[21] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J. and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849-858, Springer, Berlin.

[22] Zitzler, E. and Thiele, L. (1999). "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach". *IEEE Transactions on Evolutionary Computation*, 3(4):257-271.

[23] Zitzler, E., Laumanns, M. and Thiele, L. (2001). "SPEA2: Improving the Strength Pareto Evolutionary Algorithm". In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty, editors, *EUROGEN 2001: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 12-21, Greece, September 2001.

[24] Knowles, J. and Corne, D. (1999). "The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimisation", In Angeline, P. J.; Michalewicz, Z.; Schoenauer, M.; Yao, X. and Zalzala, A.., editors. *1999 Congress on Evolutionary Computation*, pages 98-105, IEEE Press, Washington, D.C.

[25] Knowles, J., Corne, D. and Oates, M. (2000). "The Pareto-envelope based selection algorithm for multiobjective optimization", In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J. and Schwefel, H.-P., editors, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature* (PPSN VI), pages 839-848, Springer, Berlin.

[26] Coello-Coello, C. A., Van Veldhuizen, D. A. and Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*, Kluwer Academic Publishers, New York.

[27] Coello-Coello, C. A. and Lamont, G. B. (2004). *Applications of multi-objective evolutionary algorithms*, World Scientific, Singapore.

[28] Abraham, A., Jain, L. and Goldberg, R. (2005). *Evolutionary multiobjective optimization: Theoretical advances and applications*, Springer, USA.

[29] Tan, K. C., Khor, E. F. and Lee, T. H. (2005). *Multiobjective evolutionary algorithms and applications*, Springer-Verlag, London.

[30] Wolfram, S. (1984). "Universality and complexity in cellular automata". *Physica D*, 10:1-35.

[31] Li, W. and Packard, N. (1990). "The structure of elementary cellular automata rule space". *Complex Systems*, 4:281-297.

[32] Langton, C. G. (1990). "Computation at the edge of chaos: phase transitions and emergent computation". *Physica D,* 42:12-37.

[33] Binder, P. M. (1993). "A phase diagram for elementary cellular automata". *Complex Systems*, 7:241-247.

[34] Culik II, K., Hurd, L. P. and Yu, S. (1990). "Computation-theoretic aspects of cellular automata". *Physica D*, 45:357-378.

[35] Packard, N. H. (1998). "Adaptation towards the edge of the chaos". In Kelso, J. A. S., Mandell, A. J., Shlesinger, M. F., editors, *Dynamic Patterns in Complex Systems*, pages 293-301, World Scientific, Singapore.

[36] Pagie, L. and Mitchell, M. (2002). "A comparison of evolutionary and coevolutionary search". *International Journal of Computational Intelligence and Applications*, 2(1):53-69.

[37] Morales, F., Crutchfield, J., Mitchell, M. (2001). *Parallel Computing*, 27, 571.

[38] Oliveira, G.M.B. and Siqueira, S.R.C. (2006). Parameter Characterization of Two-Dimensional Cellular Automata Rule Space. *Physica D - Nonlinear Phenomena*, v. 217, n. 1, p. 1-6.

[39] Oliveira, G. M. B., Bortot, J.C. and de Oliveira, P. P. B. (2002). "Multiobjective evolutionary search for one-dimensional cellular automata in the density classification task". In R.K. Standish, M.A. Bedau and H.A. Abass, editors. *Artificial Life VIII*, pages 202-206, MIT Press, Cambridge, MA, USA, (Complex Adaptive Systems Series).

[40] Deb, K. (1998). "Multi-objective genetic algorithms: problem difficulties and construction of test problems", *Technical Report CI-49/98*, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany

[41] Mitchell, T. *Machine Learning*. McGraw-Hill, 1997.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-160

Paraconsistent Logic Applied in Expert System for Support in Electric Transmission Systems Re-establishment

João Inácio da SILVA FILHO [a,b], Alexandre ROCCO [a], Maurício C. MÁRIO [a] and Luís Fernando P. FERRARA [a]
[a] *UNISANTA - Santa Cecilia University*
Rua Osvaldo Cruz, 266 CEP-110045- Santos – SP – Brazil
[b] *University of São Paulo - Institute For Advanced Studies*
Av. Prof. Luciano Gualberto,374 Trav.j, Térreo, Cidade Universitária
CEP 05508-900 - São Paulo –SP- Brazil

Abstract. In this work we presented an Expert System built with Paraconsistent Logic applied in a transmission electrical power system operation support in real time. The computational program forms a Paraconsistent Expert System PES capable to offer a risk analyses, diagnosis and the optimal restorative strategy proposition to the electrical power transmission system after an outage. The logic used for the PES to make decisions is the Annotated Paraconsistent Logic (APL) that belongs to a class of the Non-classic Logical denominated of Paraconsistent Logic-PL. This Paraconsistent Expert System PES uses a type of the Annotated Paraconsistent logic denominated Annotated Paraconsistent logic with annotation of two values APL2v to produce diagnosis suggesting the restorative strategy based in the analysis of occurrence information (electric Switches, Circuit breakers, protections, etc...). The use of APL brings certain advantages in comparison with the classic logic because allow to manipulate contradictory signals, and like this presenting a faster and reliable action for make decision in case of the reception of vague, ambiguous and inconsistent information. The results demonstrate that the Paraconsistent Logic, with their algorithms extracted of APL2v methodology, also opens a wide field for researches and developments and can be used with promising results for implementations of applied Expert Systems in Electric Power Systems re-establishment at different topologies.

Keywords. non-classic logic, paraconsistent logic, paraconsistent Annotated logic, distribution power systems, network distribution reconfiguration, paraconsistent analysis networks, expert system, re-establishment of electric systems.

Introduction

The electric power service interruption is considered an abnormal condition to the consumers and brings serious consequences to the society. The occurrence of defects in a electrical power system is inevitable, and the reasons that cause these interruptions are several. The outages can be internal or external to the electrical power system, or happen by the natural electric phenomena or still by human mistakes. The researches developed in this area are concentrated to find new forms of minimization of the effects of the interruptions through new techniques of restoration after the breaks partial or total of their equipments [3].

The main objective of the optimized restoration is to reduce discontinuity indexes of the supply of Electric power to the consumers. For to reach this objective a potency system should have means of to supply conditions and to be prepared for, after the occurrence, to do a discerning causes evaluation, to delimit the area with defect for the performance of the protections and to characterize the defect type. If transitory, after the elimination of the defect, the components can be put in operation by an optimized sequential way. If permanent, the part of the system that presents the defect should be identified and separate through maneuvers, and in way optimized, to make the restoration of the remaining Operation System.

All these information that should be considered in the restoration of the electrical power system can be too much stressful for the human operator that can make decisions mistaken, carting great damages. Due to the great number of information that can be conflicting or contradictory is important that the analysis and decision can be made by techniques of Artificial Intelligence, especially those that propose to applications of non-classic logics.

Many works proposing Specialist Systems for automatic re-establishment of System of Energy and substations were proposed [3] [12]. Using this reasoning line this work proposes a Expert System that uses a new methodology based in Non-classic logics. This way, this work can bring new procedures with better manner of the treatment to complex systems of conflicting information, as used in re-establishment of sub-transmission systems.

Using the theoretical concepts of the Annotated Paraconsistent Logic APL [11][12][13] we presented in this work the project of a Paraconsistent Expert System PES for analyze the possible contradictions among the received information and capable calculate the level risk and the inform the user a way optimized to make the actions for the re-establishment of Electric power Sub-system. In the analysis made by the PES the safe states and the possible contingencies are considered in which the reverse-configured system will stay stable.

The APL, as presented in [1], it allowed the manipulation of contradictory signals and the results created methods of applications through hardware and Software. In these methods presented in [9], a system that uses the Annotated Paraconsistent Logic, receives signals of information in the form of degrees of it evidences with values that vary between 0 and 1. The Systems makes the treatment of the received signals through of the algorithm and it presented in output a signal with value of certainty degree and contradiction degree.

Using these studies several researches resulted in the creation of a family of neural cells forming the Paraconsistent Artificial Neural Nets PANets. In the construction of the PES we will use three types of cells for contingency analysis, as we will see in the next sessions.

We also present a Paraconsistent Logic in a risk analysis form where is made a treatment of contradictions without invalidating the conclusions through algorithms named PANs - Para-consistent Analysis Nodes [12][14]. The PAN is formed by equations where signals are made calculations that represent information about restrictions, risks and configurations of electric power distribution systems networks. In the Figure 1 we have the diagram where the analysis pre-establishment sequence block is formed with neural artificial paraconsistent Cells and the blocks Analyze Pre-fault and analyze post-fault events is built with nets of PANs.

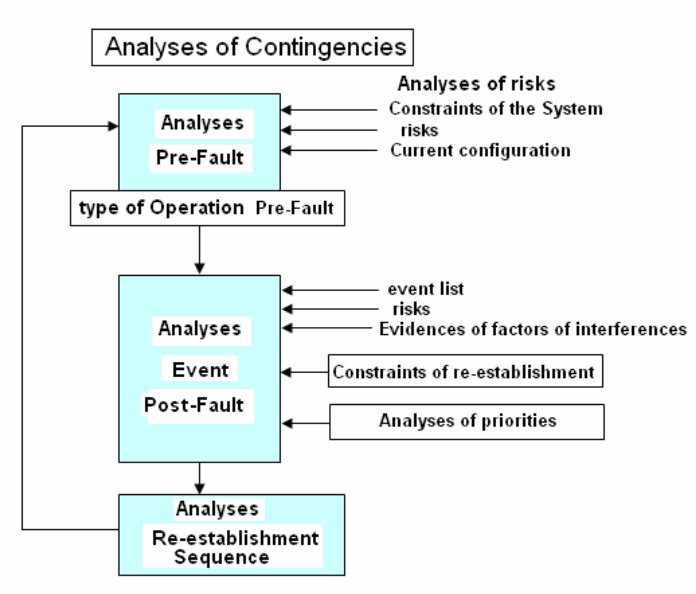


Figure 1. Flowchart of Analysis of Contingency.

The "pre-Fault state" represented by the resulting evidence Degree from PAN is evaluated together with the "post-Fault" information in a contingency evaluation. In that way the occurrence type and their parameters are classified by PANets with the purpose of offering an optimized re-establishment sequence to the power system.

The PANN can be applied in projects of Intelligent Systems and some of these Paraconsistent Expert Systems PES's have been developed and relevant works resulted in themes of master's degree theories and doctorate [10] [11] [12].

The application of the Expert System with base in a Non-classic logic makes possible the inference of data for adaptation of the maneuvers in the Electric System, conditioned to the restrictions imposed by each topology configuration of the Electrical Net and of substation. In the way are considered situations as, for instance:

1) Normal operation (parameters no violated and assisted load),
2) Operation in urgency (violated parameters and assisted load),
3) Operation in emergency (turned off load),
4) Operation in re-establishment (process of load re-establishment).

Through application of APL is possible to offer to the operator, solid information of occurrence configured by circuit breaker states, voltage and load values, protection on, as well as the profile of load of the system in real time, in a standardized way.

1. The Annotated Paraconsistent Logic with annotation of two values –APL2v

The contradictions or inconsistencies are common when we described parts of the real world. The systems of analyses, learning and recognition of the Artificial Intelligence, in general, use in his works the conventional logic, where the description of the world is considered by two states: False or True. These binary systems don't get to treat the contradictory situations generated by noises, or for lack of information on the one that wants to analyze. The Paraconsistent Logic has been created to find means of giving treatment to the contradictory situations.

The studies of the Paraconsistent Logic presented results that make possible to consider the inconsistencies [2] [4] [17], then, it is more appropriate to treat problems caused by contradictions situations that appear when we have worked with the real world.

The Annotated Paraconsistent Logic (APL) is a class of evidential logic that treats the signals represented by annotations that allows a description of real world and solves the contradictions through Algorithms. For better representation of the knowledge in treatment of uncertainties used the Annotated Paraconsistent logic denominated Annotated Paraconsistent logic with annotation of two values APL2v [7][8].

In the APL2v the proposition is accompanied with annotations. Each annotation belongs to a finite lattice and attributes a value for the correspondent proposition. We can consider that each Evidence degree is attributing to the proposition a value that belongs at the group of values composed by the constants of annotation of the lattice $\{\top, t, F, \perp\}$.

The annotation of the Annotated Paraconsistent Logic is defined through an intuitive analysis where the atomic formula p_μ is read as:

"I believe in the proposition *p* with Evidence degree μ, or until μ".

In this case, each annotated sentence for the lattice would have the following meaning:

p(t) = the sentence p is true.
p(F) = the sentence p is false.
$p(\top)$ = the sentence p is inconsistent.
$p\ (\perp)$ = the sentence p is indefinite.

This takes us to consider that the Evidence degree μ is a constant of annotation of the lattice. Each propositional sentence will come accompanied of a Evidence degree that it will attribute the connotation of "Truth", of "Falsehood", of "Inconsistency" or of "indefinite" to the proposition. Therefore, an Annotated sentence associated to the Lattice of the Annotated Paraconsistent Logic can be read in the following way:

p(t) ==> the annotation or Evidence degree *t* attributes a truth connotation to the proposition *p*.

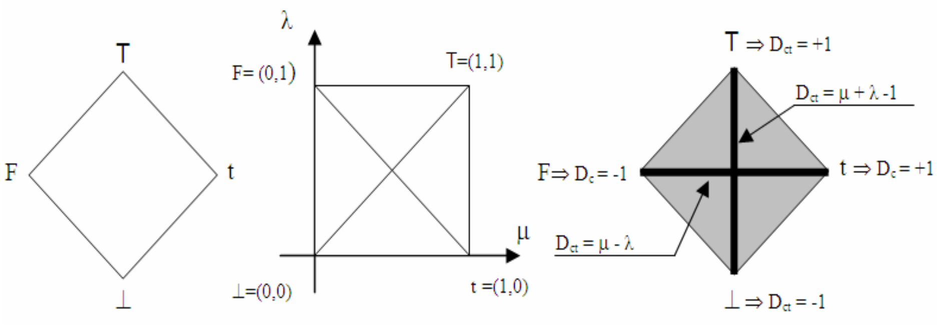
p(F) ==> the annotation or Evidence degree *F* attributes a connotation of falsehood to the proposition *p*.

$p(\top)$ ==> the annotation or Evidence degree $\top$ attributes an inconsistency connotation to the proposition *p*.

$p(\perp)$ ==>the annotation or Evidence degree $\perp$ attributes a Indefinite connotation to the proposition *p*.

Therefore, the Annotated Paraconsistent Logic with annotation of two values - APL2v is an extension of APL and it can be represented through a Lattice of four vertexes as presented in [2] where is established the terminologies and conventions. The APL2v can be represented by a Lattice (fig. 2 (a)) and to be studied through unitary square in the Cartesian plan (fig.2 (b))[7][8]. The orderly pair's first element represents the Evidence degree that is favorable at the proposition, and the second element represents the Evidence degree that is unfavorable or contrary. The second element denies or rejects the proposition. This way, the intuitive idea of the association of an annotation to a proposition $p(\mu, \lambda)$ does mean that the evidence degrees favorable at p it is μ, while the evidence degree unfavorable, or contrary at p is λ.

In the Paraconsistent analysis the main objective is to know with what value of certainty degree D_c we can affirm that a proposition is false or true. Therefore, it is considered as a result of the analysis only the value of the certainty degree D_c, and the value of the Contradiction degree D_{ct} is a indicative of the inconsistency measure. If the result of the analysis is a low certainty degree value or a high inconsistency, the result will be undefined. These values can be put in two representing axes of finite lattice, according to the fig. 2(c).



(a) (b) (c)

Figure 2. **a)** Finite lattice of APL2v four states.
b) unitary square in the Cartesian plan.
c) Finite lattice of APL2v four states with values.

The control values adjusted externally are limits that will serve as reference for analysis. A lattice description uses the values obtained by the equations results in the Algorithm denominated "Para-analyzer"[8] that can be written in reduced form, expressing a Paraconsistent Artificial Neural Cell basic PANCb. The PANCb is see in the (fig 2 (a)), and the value of the certainty degree D_c, and the value of the Contradiction degree D_{ct} can be calculated by the main equations:

$$D_{ct} = \mu + \lambda - 1 \qquad \text{and} \qquad D_c = \mu - \lambda$$

2. The Paraconsistent Artificial Neural Cells

The element capable to treat a signal composed by two degrees of evidence, where one is the Evidence favorable degree and other the Evidence unfavorable degree (μ_{1a}, μ_{2a}), supplying a output result in the form:

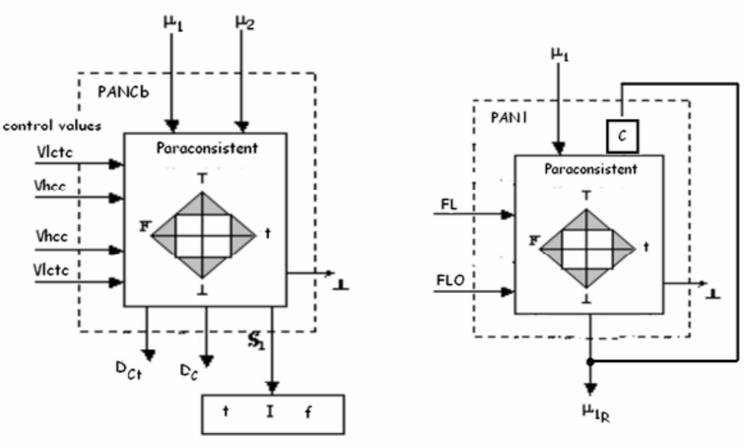
D_{Ct} = contradiction degree,

D_C = certainty degree and

X = constant of annotation indefinite, is denominated the Paraconsistent Artificial Neural Cell basic (PANCb).

The Figure 3a shows the representation of a PANCb.

The Paraconsistent Artificial Neural Cell of learning PANCL is a Paraconsistent Artificial Neural Cell basic with an output μ_1r interlinked to the input μ_{2c} (complemented disbelief degree) according to Figure 3 b.



(a) (b)

Figure 3. a)Paraconsistent Artificial Neural Cell basic PANCb.
b) Paraconsistent Artificial Neural Cell of learning (ready to receive patterns)

In the Learning Algorithm of he PANL [9], successive applied of values to the input of the evidence favorable degree (Belief Degree) (μ_1) results in the gradual increase of the evidence favorable degree or (Disbelief Degree) of the output (μ_1r). This Cell can work of two ways: by learning the truth pattern, where are applied values μ_1 =1 successively until the Belief degree of the output to arrive to the μ_1r=1, or by learning the falsehood pattern, in this last case are applied values μ_1=0 until the degree of belief resulting arrives to the μ_1r =1.

The studies of PANCb originated a family of Paraconsistent Artificial Neural Cells that constitute the basic elements of the Paraconsistent Artificial Neural Networks (PANN's). In this work, were necessary only three types [10] of Cells for the elaboration of the Paraconsistent Expert System (PES):

1-The Paraconsistent Artificial Neural Cell of Learning - PANCL.

This Cell can learn and memorize an applied pattern in its input.

2-The Paraconsistent Artificial Neural Cell of Simple Logical Connection of Maximization – PANCLs.

This Cell determines its output signal by the largest value among two applied in the input.

3-The Paraconsistent Artificial Neural Cell of Decision –PANCd.

This Cell determines the result from the Paraconsistent analysis.

3. Paraconsistent Artificial Neural Cell of learning -PANL

The Paraconsistent Artificial Neural Cell of learning PANL is a Paraconsistent Artificial Neural Cell basic with an output μ_1r interlinked to the input μ_{2c} (complemented disbelief degree) according to Figure 4.

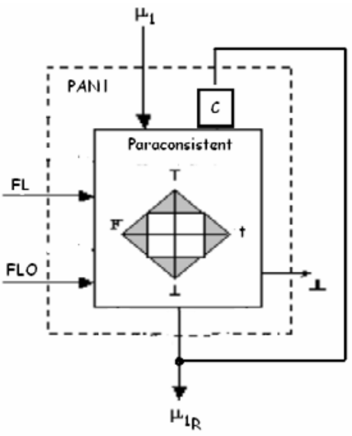


Figure 4. Paraconsistent Artificial Neural Cell of learning (ready to receive patterns)

As we see below, in the Learning Algorithm, successive applied of values to the input of the evidence favorable degree (Belief Degree) (μ_1) results in the gradual increase of the evidence favorable degree or (Disbelief Degree) of the output (μ_1r).

This Cell can work of two ways: by learning the truth pattern, where are applied values $\mu_1 =1$ successively until the Belief degree of the output to arrive to the μ_1r=1, or by learning the falsehood pattern, in this case are applied values μ_1=0 until the degree of belief resulting arrives to the μ_1r=1.

Learning algorithm for the Paraconsistent Artificial Neural Cell - PANL

```
1 - Beginning: $\mu_1 r = 1/2$                      * / virgin Cell * /
2 - Define: $FL$ = Value where: $FL \geq 1$         * / Enter with the value of the learning
                                                       Factor * /
3 - Define: $FLO$ = Value:     $FLO \geq 1$         * / Enter with the value of the Loss Factor * /
     4 - Define: $P$                                * / input Pattern, $0 \leq P \leq 1$ * /
     5 – Do: $Dci = P - \mu_{2c}$                   * / Calculates the Degree of initial belief * /
6 – If $Dci < 0$, do: $\mu_1 = 1 - P$                * / The degree of belief is the complement
                                                       of the pattern * /
7 - If $Dci > 0$, do: $\mu_1 = P$                   * / The degree of belief is the Pattern * /
8 - Do: $\mu 2 = \mu_1 r$                           * / Connects the output of the cell in the
                                                        input of the disbelief degree * /
9 – Do: $\mu 2c = 1 - \mu 2$                        * / Applies the Complement in the value
                                                       of the input of the disbelief degree * /
10 – Do: $Dc = \mu_1 - \mu 2c$                        * / Calculates the Degree of Belief * /
11 – If    $Dc \geq 0$,   do:     $C1 = FL$
```

```
12 – If    Dc < 0,    do:    C1 = FLO
13 - Do: $\mu_1 r$ = {(Dc x C1) +1} + 2          * / Found the degree of Belief resulting
                                                   in the output * /
14 - While $\mu_1 r \neq 0$,  returns to the step 8
15 - If  $\mu_1 r = 0$,  do:  $\mu_1 r = 1$ and $\mu_1 = 1 - P$   * / Applies the function NOT and it
                                                   complement the Belief degree * /
16 - Returns to the step 8
```

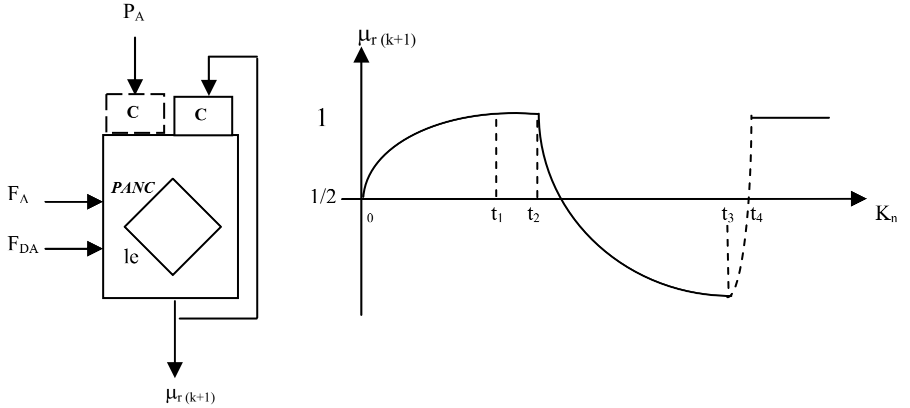


Figure 5. Simplified Representation and characteristic graph of the output signal of the Paraconsistent Artificial Neural Cell of learning PANL.

As it can be seen in the Algorithm, successive applied values to the input of the Evidence favorable degree results in the gradual increase in the resulting Evidence degree $\mu_1 r$ from the output. This Cell can work of two manners:

1) Learning of the truth pattern, where they are applied values =1 successively until that the resulting Evidence degree in the output arrives to $\mu_1 r = 1$.

2) Learning of the falsehood pattern where are applied values = 0 until the resulting Evidence degree to arrive to $\mu_{1r} = 1$, in this case the input of the Evidence favorable degree μ_c is complemented.

4. Paraconsistent Analyzer Node - PAN

In [8] a method for the treatment of uncertainties using Annotated Para-consistent Logic is presented. Two values are considered as outputs of the analysis:

A real certainty Degree D_{Cr} calculation, for:

$$D_{Cr} = 1 - \sqrt{(1 - |D_C|)^2 + D_{ct}^2}$$

If: $D_C > 0$

and:

$$D_{Cr} = \sqrt{(1-|D_C|)^2 + D_{ct}^{\ 2}} - 1$$

If: $D_C < 0$

and an Interval of Certainty $\varphi_{(\pm)}$ for:

$\varphi = 1 - |D_{ct}|$

Where: $\varphi = \varphi_{(+)}$ If $D_{ct} > 0$

$\varphi = \varphi_{(-)}$ If $D_{ct} < 0$

According to [8] an Algorithm that makes this type of analysis is named System or Paraconsistent Analysis Node PAN.

A PAN is capable of receiving evidences and supplying a certainty value accompanied of its Interval of Certainty. Therefore, it is considered a Para-consistent Analysis Node – PAN the System of Analysis that receives Evidence Degrees in their inputs and supplies two values; one that represents the real Certainty Degree D_{Cr} and another, that is the signal of the Interval of Certainty $\varphi_{(\pm)}$.

5. Algorithm of the Paraconsistent Analyzer Node

With the considerations presented here we can compute values using the obtained equations and build a System of Para-consistent analysis capable of offering a satisfactory answer derived from information collected from uncertain knowledge data base. The PAN-Paraconsistent Analyzer Node is built by the "Algorithm of Paraconsistente Analysis of APL2v"as described bellow.

```
1. Enter with the input values
        $\mu$ */ favorable evidence Degree    $0 \le \mu \le 1$
        $\lambda$ */ unfavorable evidence Degree $0 \le \lambda \le 1$
2. Calculate the Contradiction Degree
        $D_{ct} = (\mu + \lambda) - 1$
3. Calculate the Interval of Certainty
        $\varphi = 1 - |D_{ct}|$
5. Calculate the Certainty Degree
      $D_C = \mu - \lambda$
6. Calculate the distance D
      $D = \sqrt{(1-|D_C|)^2 + D_{ct}^{\ 2}}$
4. Determine the output signal
      If  $\varphi \le 0,25$ or $D > 1$ then do $S1 = 0.5$ and $S2 = \varphi$:
        Indefinite and
        go to the item 10
                                    Or else go to the next step
7. Calculate the real Certainty Degree
      Se $D_C > 0$      $D_{Cr} = (1 - D)$
      Se $D_C < 0$      $D_{Cr} = (D - 1)$
8. Determine the signaling of the Interval of Certainty
        If  $\mu + \lambda > 1$   Signal positive $\varphi_{(\pm)} = \varphi_{(-)}$
        If  $\mu + \lambda < 1$   Signal negative $\varphi_{(\pm)} = \varphi_{(+)}$
        If $\mu + \lambda = 1$   Signal zero   $\varphi_{(\pm)} = \varphi_{(0)}$
9. Present the outputs
        Do S1 = $D_{Cr}$   and   S2= $\varphi_{(\pm)}$
10. End
```

Two more lines are brought into the algorithm If there are connections among PANs forming networks of para-consistent analysis.

```
   9. Calculate the real Evidence Degree
        $\mu_{Er} = \frac{1+D_{Cr}}{2}$
   10. Present the outputs
        Do  S1 = $\mu_{Er}$ and  S2= $\varphi_{(\pm)}$
   11. End
```

The symbolic representation of a PAN [12][13]is presented in Figure 6 where we have two inputs; favorable Evidence Degree μ and unfavorable Evidence Degree λ of the regarding analyzed Proposition *p* and two output signals of results; the real Certainty Degree D_{cr} and the Interval of Certainty symbolized by $\varphi_{(\pm)}$.

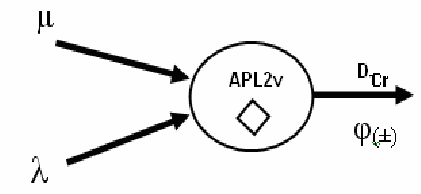


Figure 6. Symbol of the PAN - Paraconsistent Analyzer Node.

The application of Paraconsistent Logic through the methodology of APL2v presented in [4] considers propositions and works in an evidential mode. Thus, creating propositions the Degrees of evidences that will feed the PANs is modeled by extracting information from measuring points, heuristics and data base.

6. Analyses of Contingencies with Risks Identification

The APL2v accepts extracted signals of evidences of contradictory information. With Paraconsistent Logic application is possible the inference of data for analysis of pre-fault states and his comparison with the post-fault state. In that way it is possible that, through the results of the analysis, an adaptation of the maneuvers is applied for the re-establishment of the electric power system. These maneuvers are directly conditioned to the topologic configuration of the substation and network. It is then considered that an Expert System should make control actions in three states of analysis to act in support for the re-establishment of electric power systems[12][14].

1. **Pre-fault** – Analyze of the System in operation.

2. **Post-Fault** - Analyze of the System in the contingency.

3-**Re-establishment** – Analyze of the System after contingency.

In a distribution system these three states are in a continuous loop of analysis and actions. The ideal is that the system always stayed in the state of pre-fault analysis. For each one of these states a Paraconsistent Analysis Network PANet composed of interlinked PANs makes the analysis generating evidences that will allow the re-establishment of the electric power system.

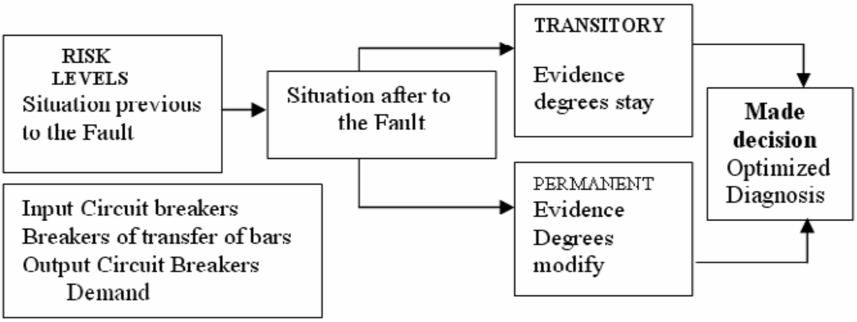


Figure 7. Actions and analysis of a Expert System of Re-establishment of Electric power System.

The PAN aiming at a great plan for re-establishment that should satisfy the following items:

a) to find a plan in a short interval of time (real time).

b) to minimize the number of maneuvers.

c)to recompose the System in the closer type of Operation possible of the state Pre-fault.

d)to reduce the number of interrupted consumers.

e) to assist the priority consumers.

f) to make arrangements so that no component is overloaded.

g) to maintain the radial structure of the System (without formation of rings).

h) Other objectives depending on the need of the company.

7. Composition of the Paraconsistente Analysis Network-PANet for identification Risks

In this work we will focus on the Paraconsistent analysis in the actions of the state of Pre-Fault. The Paraconsistent analysis in this state of Operation will originate the conditions to, along with other factors; form a sequential closing of breakers for the re-establishment of the System of Sub-transmission of Electric power in the post-fault state.

The ALP2v methods applied to the pre-fault actions are:

1. **Pre-Fault State** – In that state the System is in operation regime.

The System should be capable of analyzing and classifying the operation type. The type of Operation can be classified for instance, as one of the presented in [5]:

a) Normal operation (parameters not violated and assisted load)
b) Operation in urgency (violated parameters but assisted load)
c) Operation in emergency (load off)
d) Operation in restoration (process of load re-establishment).

The classification done by the Paraconsistent Analysis Network PANet will just generate an evidence signal whose value will define the operation type through a single proposition object (Po).

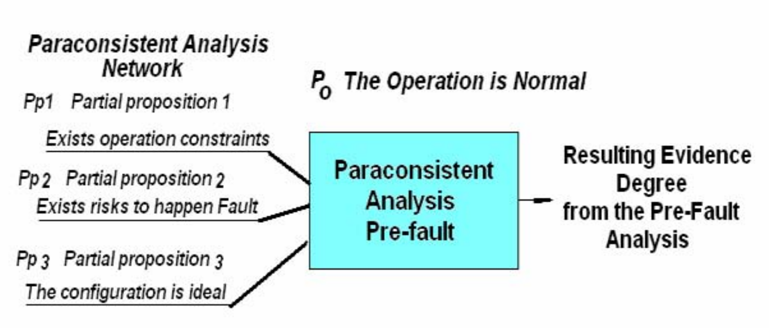

Paraconsistent Analysis Network
P_O The Operation is Normal
Pp1 Partial proposition 1
Exists operation constraints
Pp2 Partial proposition 2
Exists risks to happen Fault
Pp3 Partial proposition 3
The configuration is ideal
Paraconsistent Analysis Pre-fault
Resulting Evidence Degree from the Pre-Fault Analysis

Figure 8. Paraconsistent Analysis in Pre-fault operation state

When the resulting Evidence Degree reaches the value 1 it means that the analyzed evidences acted by the partial propositions are confirming the object proposition. When the value of the resulting Evidence Degree decreasing and approaches 0.5 (the Indefinite state), it means that the information brings forward evidence that weakens the affirmative to the proposition. In these conditions the analysis indicates that some parameters are violated in spite of the assisted load. An investigation in the PANs about the values of the evidence degrees of the partial propositions and their evidence intervals, allows an indication of the origin of the violation of the parameters and of the contradictions that are provoking that decrease of the resulting Evidence Degree from the object proposition.

When the resulting Evidence Degree crosses the value of the indefinite state 0.5 and approaches zero it means that the information that brings forward the evidences for the analysis about the partial propositions are indicating a larger refutation to the proposition object. Therefore, the evidences of risks, related to the restrictions and the current configuration of the system suggest that it is approaching an Emergency Operation State. investigation in PANs about the values of the evidence degrees of the partial propositions and their evidence intervals, brings information that qualify the formation of a better action in the sense of increasing the Degree of Evidence of the proposition object, to take it to the maximum value 1. Therefore, to take the Power System for the state of normal operation and without risks.

8. The Paraconsistente Analysis Network-PANet

According to the fundamental concepts of the Paraconsistent Logic an analysis should admit contradictions. This means that, when receives contradictory information the Paraconsistent System analyzes them and, without allowing the weight of the conflict to invalidate the analysis, he always produces an answer. It produces a value that expresses the reality. The linked PANs in the Paraconsistent Analysis Networks PANet are extracted algorithms of the APL2v and, unlike other types of treatment of uncertainty they do not admit factors, weight or changes in their structure that can compensate types of evidences of their inputs. For that reason the evidence Degrees presented for analysis should express the nature and the characteristics of the source of information. And because of that models are made and the variations within the discourse universe as well as the interrelation with other sources of information are considered.

9. Modeling of the signals of evidence degrees inputs of System for identification Risks

After the choice of a Proposition all of the possibly available evidences that will help to affirm that proposition will be found through knowledge extraction. The regarding Evidence Degrees will be modeled in the Discourse universe and with a variation that will depend on the nature of the source of information.

Other sources of information that will supply the Evidence Degrees for PANs to make partial proposition analysis are for instance; the signals originated from the System SCADA, that bring measurements of the tension, current e loads, rele states and protection, besides the profile of load of the system in the real time and topology of System.

The modeling and the extraction of information with the objective of generating the degrees of evidences to the risks are made in several ways as; using heuristics, it seeks in databases, interpretations of linguistic variables, statistical calculations, etc.

9.1. Modeling of the risks

The risks can be classified and normalized from that classification,. Then they are transformed into Evidences Degrees to be used in the analysis of the PANet .

In this a classification of risks for analysis of contingency was made in the following way:

Risks of the Switching P_{sh} – They are related to the real configuration of the System of electric power Sub-transmission. The configuration is related to the actual Switching of the distribution system configured by breakers states and electric Switch in the real topology of the system. The proposition object of each bus and path circuits is related to the constraint on states of the breakers. Therefore, it is of the type: P_{sh} = The state of the breaker B_n is ON.

Non-controllable risks P_{NC} – They are the ones related to the nature: schedule of pick, incidence of rays, day of production, etc...

Controllable risks P_C – They are the ones related to the electric measurements as: Current, tension, flow of loads, etc.

10. Topology of The Paraconsistent Analysis Network-PANet For Risk Identification

The PANs are interlinked in PANet with their own modelling for each specified analysis of each area of the System of electric power Sub-transmission. Figure 9 shows the PANs interlinked for risk analysis used in this work.

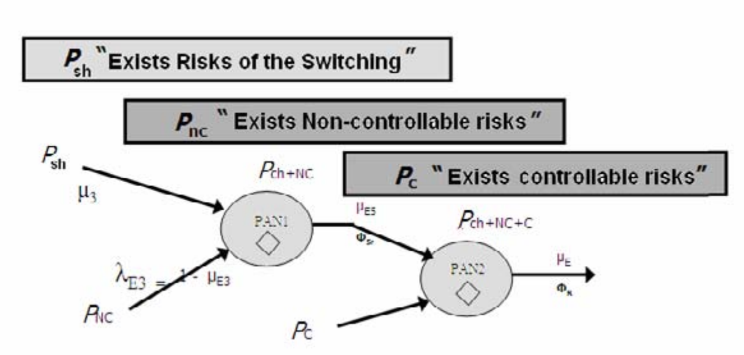

P_{sh} "Exists Risks of the Switching"
P_{nc} " Exists Non-controllable risks"
P_c " Exists controllable risks"
P_{sh}
μ_3
P_{ch+NC}
PAN1
μ_{E3}
Φ_{3r}
$P_{ch+NC+C}$
PAN2
μ_E
Φ_R
$\lambda_{E3} = 1 - \mu_{E3}$
P_{NC}
P_C

Figure 9. PANs interlinking for risk analysis

Using the methodology and the equations of the APL2v, a high resulting evidence degree from the constraints of the partial proposition (*P*) result in a low evidence Degree of the proposition *P*o. The value of the resulting Evidence Degree obtained in the output of PANet that analyzes the object Proposition: "The System is Normal" indicates what is the type of operation of the System and it will be Evidence for the analysis made by the network in the Post-Fault condition. The topology of the network of Analysis used in this work is presented in Figure 10.

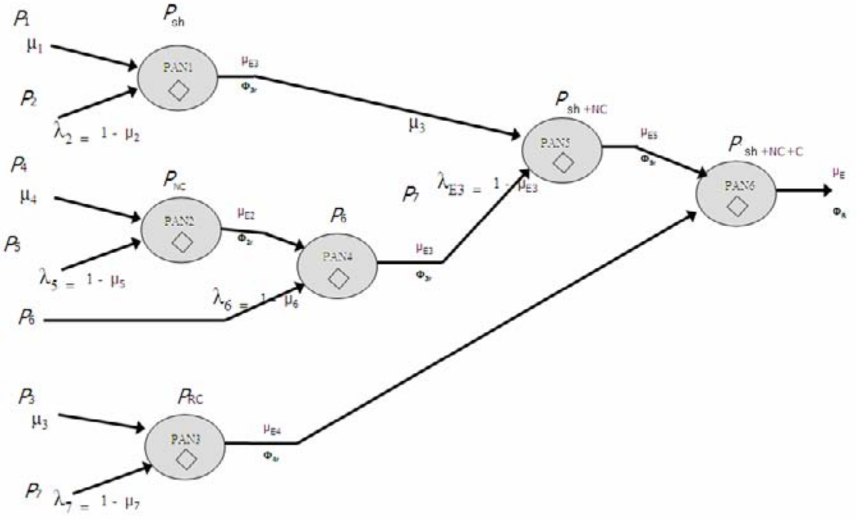


Figure 10-. Interlinked PANs in a Paraconsistent Analysis Network for analysis of Risk Evidence in a bus of the Power System.

We explained that other types of topologies can be used. Different topologies depend on the analysis characteristics and the nature of the sources of information used as generators of evidence degrees to feed the network.

In this design that treats of analysis of contingencies in the substation of Electric power feeds the PANet with information on the switching state and load in real time. At the same time in that it is generated the information the substation is monitored by the analyzer of risks. It is verified in that the substation operation type is. Are the states of the switches and of the breakers of the system, together with the measurements of the loads in real time of the substation that provide information about Electric System configuration.

These information that, as it was seen, they enter in the classification of risks for switching, are analyzed in real time and stored temporarily. In the occurrence of the fault in the electric power System the analysis done by PANet change for Pos-fault state. In this case, the information that were stored will be used as evidences, together with the value of the Evidence Degree of the type of fault Operation that happened and will be capable by an paraconsistent analysis, to offer a suggestion of better sequence for the re-establishment.

11. The Paraconsistent Expert System for Electric Power System Re-establishment

For the Electric power System re-establishment in an efficient and fast way the Expert System should analyze all of the information and to make the decisions for turn on the electric breakers and reinstating the system for a possible closer configuration before the occurrence of the contingency. It should also make the exclusion of the components that were affected for the permanent fault.

This re-establishment optimized is constituted in sequences of operations considered correct and are necessary the analyses of information after and before the defect. These information and others from the electric circuits can increase the complexity of the data that a Expert System should to treat to make the decision.

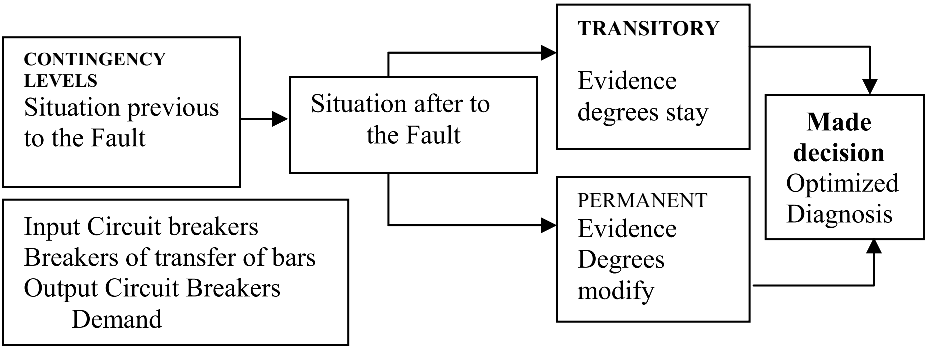


Figure 11. Actions and analysis of a Expert System of Re-establishment of Electric power System.

We considered initially that in previous events the procedures and norms adopted by the human operator when in the occurrences of contingencies are registered inside certain techniques of re-establishments. The Paraconsistent Expert System PES uses this information, as a file of studied cases. The attitude or procedures of the human operator based on these rules (that can be subjective form, vacancies or contradictory) receive a valuation for treatment with base in the Annotated Paraconsistent Logic APL.

It is made a confrontation with diagnoses, and actions registered along the re-establishments of the Electric power System, and a validation of the operative actions through the Expert System built with the Paraconsistent Artificial Neural Networks Artificial PANN's.

In this first phase the optimized diagnosis of PES will be compared with an action of a human Operator in the contingency represented by a word. For example, with the values of the Evidence degrees related to the propositions of the Sub-system, is formed a word composed with all the involved electric Switches. Joining the values of the restrictions and optimized diagnosis, the Paraconsistent word is in the following form:

Before-contingency Conditions	Re-establishment
B1 B2 B3 B4 B5 B6 B7 B8 .... Bn $X\mu_a$ $X\mu_b$ $X\mu_A$ μ_{Res}	B4 B7 B8 B... Bn μ_{RO} μ_{RD}

Where: B1 B2 … Bn = Evidence Degrees of the Circuit Breakers of the Input, of the electric power transformation and of the Output.

$X\mu_a$ $X\mu_b$ $X\mu_n$ = Evidence Degrees of the Demand

μ_{Res} = Evidence Degree of the Maximum restriction

μ_{RD} = Evidence Degree of the Optimized diagnosis

For the end of the analysis, the proposition that will be annotated by the Evidence degree resultant will be: "The re-establishment of the Sub-system of Electric power is optimized".

Other types of re-establishment of the Sub-system are classified receiving each one the values regarding the Evidence degrees compared to the optimized.

With the PES these Circuit Breakers will compose the object of decision; therefore, the decision will be made in the sense of, after the paraconsistent analysis and the obtaining of the diagnosis, the actions of re-establishment of the System and the correspondents' maneuvers of this group of electric Switches be showed.

The diagram in blocks of the Figure 12 shows the configuration of a PES.

The PES, through the Paraconsistent Artificial Neural Networks treats the signals represented by the Evidence degrees and compares them with results learned previously by the training of the net. In the end of the analysis, the PES presents the re-establishment of the Sub-system in an optimized way. Depending on the result, a cast of suggestions will be supplied for the re-establishment of the Sub-system of Electric power.

In this work they are considered 4 types of Re-establishment of the Sub-system of Electric power. They are: Optimized re-establishment, Correct re-establishment, Minimum re-establishment and Re-establishment Not Correct.

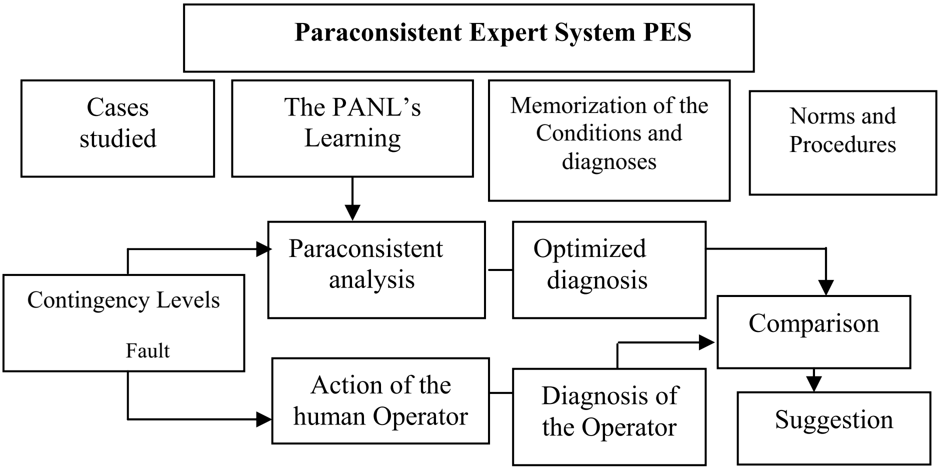


Figure 12. Flowchart of the Paraconsistent Expert System PES

12. Architecture of the Paraconsistent System formed with Neural Artificial Paraconsistent Cells

The Paraconsistent System with neural artificial paraconsistent Cells to the to the Electric transmission system operation support is composed by modules denominated: Paraconsistent Artificial Neural Unit of Comparison of Words - PANUCW.

The PANUCW store Words patterns that will be compared with others that will be applied in the inputs. Each Unit - PANUCW is composed by a group of Paraconsistent Artificial Neural cells that compose a system of learning of Paraconsistent words PANSls.

The PANSls are used as sub-nets (wafers) that have inside for purpose to separate the words that present the several similar patterns of the Paraconsistent Artificial Neural Networks. In his operation all the patterns regarding the character of the certain Paraconsistent word (PA1, PA2,..., PAn) are stored in a same PANSls system.

When this system is stimulated by an external pattern (μ_2), he makes the comparison of this Word with stored them and making a Paraconsistent analysis. In the output of the PANSls the pattern that more resembled at stored signals is presented.

The diagram blocks from the Figure 13 shows how a Paraconsistent Expert System is composed. The PANCW Modules can be joined to expand the recognition capacity. The amount of interlinked modules will depend of project application.

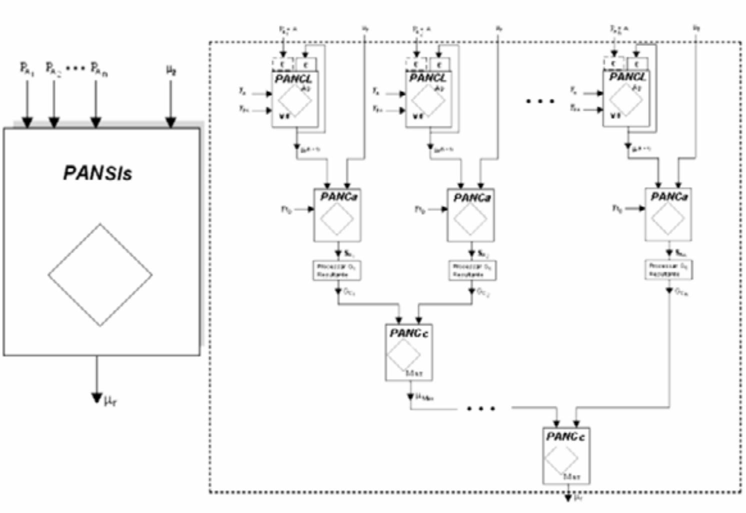
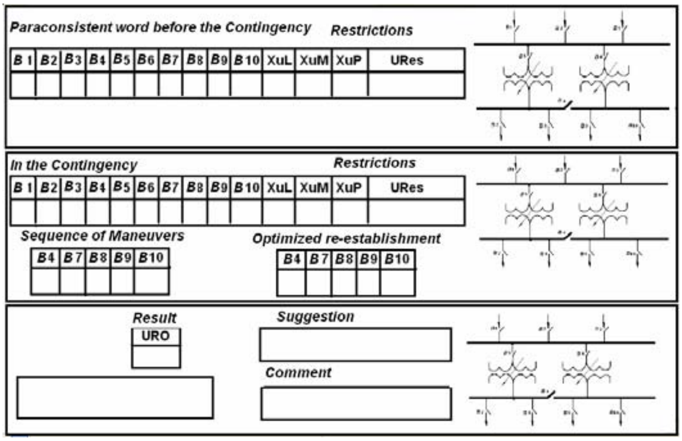


Figure 13. Representative Block of the PANSls and Structures of the PES

The computational program of the PES was implemented in VB (Visual Basic) and his main window is shown in the Figure 14.

The main screen of the PES was built so that after the learning certain number of Paraconsistent words a certain contingency types is introduced together with the action of the human Operator to make the re-establishment of the Sub-system.



Paraconsistent word before the Contingency Restrictions

B 1	B2	B3	B4	B5	B6	B7	B8	B9	B10	XuL	XuM	XuP	URes

In the Contingency Restrictions

B 1	B2	B3	B4	B5	B6	B7	B8	B9	B10	XuL	XuM	XuP	URes

Sequence of Maneuvers

B4	B7	B8	B9	B10

Optimized re-establishment

B4	B7	B8	B9	B10

Result

URO

Suggestion

Comment

Figure 14. The main screen of the PES

After the recognition is exhibited, in the output of the network, the pattern that it identifies the type of Re-establishment and the Evidence Degree resulting of the analysis came with suggestions to optimize the re-establishment Electric System. Still for a better evaluation of the experiment, it is also exhibited the Evidence Degree resulting from the three Paraconsistent words that, in the analysis, are more near of the Optimized Recognition.

13. Practical results of the System formed with neural artificial paraconsistent Cells

Initially, the network was trained to learn one serializes of Paraconsistent words where the Evidence degrees presented differences of 0.25 amongst themselves. To follow words were inserted Evidence degrees with closer values hindering the capacity of the network to recognize and to establish classification among different types of formats of words. Same in these conditions the PES presented a good recognition of the defects and manners of re-establishment of Sub-systems of Electric power. However the great advantage that the Paraconsistent Expert System presented was the easiness of insert new cases for analysis and the good conditions of adjustments for the purification and choice of the cases for the reach an Optimized re-establishment.

14. Conclusion

In this paper we present a Contingency Analyzer that makes the analysis of risks and proposition of the optimal restorative strategy to the electrical power transmission system after an outage based on Annotated Paraconsistent Logic. The study of the project is always made in the reasoning line used in Artificial Intelligence which allows us to show the several sources of information that compose a electric power system. The design of the analyzer of contingencies demanded interpretative efforts in the extraction of the knowledge and in the methodology of application of the Paraconsistent Logic, which considers all of the sources as generators of signals in the form of Evidence Degrees. It is verified that these information, extracted and modeled are appropriate for treatment of signals using PANs - Paraconsistent Analysis Nodes. The PANs and Paraconsistent Artificial Neural Cells interlinked in network are capable of making an analysis through evidences of risks and configuration of switches of the System in real time. In that way the PANs generate information in the form of resulting Evidence Degree that, in the fault occurrence, will be used as information for the re-establishment of the System by the net composed of the Paraconsistent Artificial Neural Cells. The contingency analysis presented in this work should be considered as a small part of a great Paraconsistent expert system PES that, through APL, answers in a closer way to the human reasoning. The paraconsistent risk analyzer and contingency analyzer are being studied in an off-line system applied to a small pilot System of Sub-transmission of Electric power composed by 2 buses and a substation of small load. The contingency analyzer has presented good results and answers well to several situations when compared to the answers to previous situations, memorized in databases. The modulation parameters are of easy adjustment and the analyzer of contingencies is easily adapted to present resulting information suitable with reality. The next step is to adapt the resulting evidence Degrees of the contingency analysis to suggest re-establishments that select the several possibilities of maneuvers for transfers of loads.

References

[1] ABE, J.M., Fundamentos da Lógica Anotada, Tese de Doutorado, FFLCH - USP, 135 pp, 1992.

[2] ABE, J.M. & J.I. DA SILVA FILHO, Inconsistency and Electronic Circuits, *Proceedings of The International ICSC Symposium on Engineering of Intelligent Systems* (EIS'98), Volume 3, Artificial Intelligence, Editor: E. Alpaydin, ICSC Academic Press International Computer Science Conventions Canada/Switzerland, ISBN 3-906454-12-6, 191-197, 1998.

[3] CIGRE, Pratical use of expert systems in planning and operation of power systems, TF 38.06.03, Électra, n.146, pp30-67, fevereiro 1993

[4] DA COSTA, N.C.A. & J.M. ABE, Aspectos Sobre Aplicações dos Sistemas Paraconsistentes, Atas do I Congresso de Lógica Aplicada à Tecnologia – LAPTEC'2000, Editôra Plêiade, São Paulo, SP – Brasil, Editor: J.M. Abe, ISBN 85-85795-29-8, 559-571, 2000.

[5] DA COSTA, N.C.A., J.M. ABE, J.I. DA SILVA FILHO, A.C. MUROLO & C.F.S. LEITE, Lógica Paraconsistente Aplicada, ISBN 85-224-2218-4, Editôra Atlas, 214 págs., 1999.

[6] DA COSTA, N.C.A., J.M. ABE & V.S. SUBRAHMANIAN, Remarks on annotated logic, *Zeitschrift f. math. Logik und Grundlagen d. Math*. 37, pp 561-570, 1991.

[7] DA SILVA FILHO, J.I., Implementação de circuitos lógicos fundamentados em uma classe de Lógicas Paraconsistentes Anotadas, Dissertação de Mestrado-EPUSP, São Paulo, 1997.

[8] DA SILVA FILHO, J.I., Métodos de interpretação da Lógica Paraconsistente Anotada com anotação com dois valores LPA2v com construção de Algoritmo e implementação de Circuitos Eletrônicos, EPUSP, Tese de Doutoramento, São Paulo, 1999.

[9] DA SILVA FILHO, J.I.& ABE, J.M. *Fundamentos das Redes Neurais Artificiais - destacando aplicações em Neurocomputação.* 1.ed. São Paulo, Editora Villipress, Brazil 2001.

[10] DA SILVA FILHO J.I., Rocco, A, Mario, M. C. Ferrara, L.F. P. "Annotated Paraconsistent logic applied to an expert System Dedicated for supporting in an Electric Power Transmission Systems Re-Establishment" IEEE Power Engineering Society - PSC 2006 Power System Conference and Exposition pp. 2212-2220, ISBN-1- 4244-0178-X – Atlanta USA – 2006.

[11] DA SILVA FILHO J.I., Santos, B. R. M., Holms, G. A. T. A., Rocco A. "The Parahyper Analyzer: A Software Built With Paraconsistent Logic To Aid Diagnosis Of Cardiovascular Diseases" Proceedings "Safety, Health and Environmental World Congress" SHEWC 2007, July 22-25, 2007. Published by Claudio da Rocha Brito (ISBN 85-89120-47-3) & Melany M. Ciampi (ISBN 85-89549-43-7), Santos-SP Brazil 2007.

[12] DA SILVA FILHO J.I. , Rocco A., Mario, M.C., Ferrara L. F. P. "PES- Paraconsistent Expert System: A Computational Program for Support in Re-Establishment of The Electric Transmission Systems" Proceedings "VI Congress of Logic Applied to Technology" LAPTEC2007 p.217, ISBN 978-85-99561-45-4 - Santos / SP / BRAZIL - November 21-23, 2007.

[13] DA SILVA FILHO J.I. Rocco A., Onuki A. S., Ferrara L. F. P. and Camargo J. M. "Electric Power Systems Contingencies Analysis by Paraconsistent Logic Application" 14th International Conference on Intelligent System Applications to Power Systems (ISAP2007) November 4-8, pp 112-117- kaohsiung, Taiwan, 2007.

[14] FERRARA, L.F.P. *Redes Neurais Artificiais Aplicada em um Reconhecedor de Caracteres* Dissertação de Mestrado - UFU, Uberlândia-MG, 2003.

[15] MARIO, M. C, Proposta de Aplicação das Redes Neurais Artificiais Paraconsistentes como Classificador de Sinais utilizando Aproximação Funcional - Dissertação de Mestrado - UFU, Uberlândia-MG, 2003.

[16] MARTINS, H.G., *A Lógica Paraconsistente Anotada de Quatro Valores-LPA4v Aplicada em um Sistema de Raciocínio Baseado em Casos para o Restabelecimento de Subestações Elétricas* –UNIFEI – Tese de Doutoramento Itajubá,MG, 2003.

[17] SUBRAHMANIAN, V.S "On the semantics of quantitative Lógic programs" Proc. 4 th. IEEE Symposium on Logic Programming, Computer Society press, Washington D.C, 1987.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-180

Fuzzy Dynamical Model of Epidemic Spreading Taking into Account the Uncertainties in Individual Infectivity

Fabiano de Sant'Ana dos SANTOS [a], Neli Regina Siqueira ORTEGA [b], Dirce Maria Trevisan ZANETTA [a] and Eduardo MASSAD [b]

[a] *Department of Epidemiology and Public Health, School of Medicine of São José do Rio Preto*
Av. Brigadeiro Faria Lima 5.416, CEP: 15090-000, São José do Rio Preto – SP, Brazil
Phone:+55 17 32105740
[b] *Medical Informatics, School of Medicine of University of São Paulo,*
Rua Teodoro Sampaio 115, Pinheiros, CEP: 05405-000, São Paulo – SP,
Brazil - Phone: +55 11 30617682 - Fax: +55 11 30617382

Abstract - In this paper we present a fuzzy approach to the Reed-Frost model for epidemic spreading taking into account uncertainties in the diagnostic of the infection. The heterogeneities in the infected group is based on the clinical signals of the individuals (symptoms, laboratorial exams, medical findings, etc.), which are incorporated to the dynamic of the epidemic. The infectivity level is time-varying and the classification of the individuals is performed through fuzzy relations. Simulations considering a real problem data of the influenza epidemic in the baby daycare are performed and the results are compared with a stochastic Reed-Frost generalization developed by the authors in a previous work.

Keywords: Fuzzy logic, Fuzzy epidemic, Reed-Frost, Epidemiological models.

Introduction

Fuzzy dynamical systems still consist in a challenging area, particularly for the modeling of non-linear systems. Models based on differential equations have been proposed by several authors [1-2]. However, these approaches are difficult to apply in epidemiology due to the fact that epidemic models have, in general, strong non-linearity. In order to incorporate the heterogeneities in ecological and epidemiological models, Barros *et al.* considered fuzzy parameters in the differential equations [3-5], which solution could be found by calculating the *Fuzzy Expected Value* whenever the variables have a probabilistic distribution. Although these models consist in a significant contribution to the field, applying it is not an easy task. An alternative approach for fuzzy epidemic models based on dynamical linguistic models has been proposed [6-7]. However, these gradual rules systems present important limitations, as the explosion of the number of the rules and the difficulties of the experts to model the consequents if many input variables are considered [7]. In this context, any approach of fuzzy dynamic model applied in epidemiology may be an important contribution for both fuzzy and epidemic areas.

Epidemic systems are, in general, described for the most part through macroscopic models, in which the dynamic is based on the population parameters such as force of infection, mortality rate and recover rate. On the other hand, there are few microscopical epidemic models available [5,8], that is, models whose individuals` information affect the population dynamic.

Many models of epidemic spreading have been proposed to help in the comprehension of infectious diseases, with the obvious assumption that knowledge could help in the control of these diseases [9-10]. The simplest macroscopic epidemic model available in the literature is the called Reed-Frost model.

The Reed-Frost model was proposed by L. J. Reed and W. H. Frost in a series of lectures held at Johns Hopkins University [11-12]. It is a particular case of a chain-binomial model, in which it is assumed that each infected individual infects susceptible individuals independently, and that individuals are under the same contact rate with each other. If we represent by p the probability of a contact between a susceptible and an infected individual resulting in a new case, we have that, at time t, the probability that a susceptible individual does become infected, Ct, is equal to the probability of at least one infectious contact, that is,

$$C_t = 1-(1-p)^{I_t}, \qquad t>0, \tag{1}$$

where it is equal to the number of infected individuals at time t. Time t is assumed to be a discrete variable, and an individual's classification, as either susceptible, infected or resistant, can only change when time changes from t to t+1.

In the Reed-Frost model it is assumed that the probability of an infectious contact is fixed along the epidemic course and it is the same for all individuals. Therefore, neither heterogeneities in the both susceptible and infected groups nor errors involved in the classification process are considered. Due to its assumptions, the Reed-Frost model is adequate to describe infectious diseases that spread in closed and uniformly-mixed groups, whose size N is constant and small. However, the homogeneity assumption does not hold in a majority of real epidemics, since each individual may present different susceptibility and infectivity levels, depending on environmental, physiological and psychological factors. In certain cases, the assumption of time-invariant susceptibility/infectivity levels does not hold either.

In addition, errors in the diagnosis process are likely for a great number of infectious diseases, especially when the diagnostic test is neither readily nor easily available, as in the case of dengue, influenza and several other viral and bacterial infections. In those cases, the diagnostic process involves uncertainties, and is usually based upon a set of clinical characteristics, often subjective and vague, which we call signals. Indeed, the infectivity level of an infected individual may depend upon the set of signals developed.

There have been several attempts to generalize the Reed-Frost model so as to consider a non-homogeneous group, either from the susceptibility or from the infectivity points of view [12-15]. In all these, the homogeneity assumption is relaxed by dividing the main group into subgroups, and considering that there is homogeneous mixing. Subgroups are closed and individuals remain within the same subgroup for the entire duration of the epidemics, which means that an individual's susceptibility and infectivity levels are taken as constant throughout the epidemic course. However, it would be interesting if the individual's heterogeneities were treated without the separation of the population in subgroups, incorporating it directly into the dynamics of the system.

An interesting Reed-Frost generalization was proposed by Menezes and collaborators [8], that consider the clinical signals involved in the classification process in the study of the epidemic course. Those clinical signals may include symptoms, results from laboratory and physical examinations. They assumed that, after being infected, no resistance is gained and the individual becomes susceptible again (Susceptible-Infected-Susceptible model). The individual's infectivity is modeled as a function of the signals, therefore allowing for time-dependent, heterogeneous infectivity. In this work susceptibility levels are kept constant. Since this model involves only random variables it is possible to obtain important epidemical expressions, such as the epidemic basic reproduction number and its probability function [8].

In this paper, we propose a generalization of the classical Reed-Frost model, including the clinical signals presented by each individual of the group in the classification process, through a fuzzy decision process. By doing this, we intend to incorporate individual heterogeneities in the classificatory process: signals are used to define whether an individual is infected or susceptible, and also to define how the epidemics will spread.

1 A Reed Frost Model with Heterogeneities

In order to consider the individual's heterogeneities in the Reed-Frost model, in terms of infectivity or susceptibility, several generalizations were proposed [13-14]. These attempts consist in macroscopic epidemic models. In contrast, the Menezes et al. [8] Reed-Frost generalization consists in a microscopic model, in which the uncertainty involved in the diagnostic classification is modeled through a stochastic process and based on individual's information.

In the model proposed by Menezes et al. the clinical signals are recorded and are taken into account in the epidemic course via a signal summary, both as part of the classification process and to define the probability of an infectious contact [8]. It is assumed that, the higher the signal summary, the higher the probability that a contact be infectious. The probability that an individual has at least one infectious contact, which is the core of the Reed-Frost model, is then computed taking into account the heterogeneous infectivity in the group.

The model can include both signals linked to an increased infectiousness and signals linked to a decreased infectiousness. Both types of signals enter the signal summary, affecting it in opposite directions. Distinct signals can have different weights in the summary, reflecting the impact they are believed to have on both the classification process and on the infectious contact probability.

A probability distribution is assigned to the signal summary, conditioning on the previous probability of at least one infectious contact. This distribution is a mixture of the one given the individual is infected, with the one given the individual is susceptible. In this approach the classification is seen as a probabilistic step conditioned on the signal summary. The probability of an infectious contact is taken as a deterministic, polynomial function of the signal summary.

In this formulation a generalized Reed-Frost model is constructed taking a susceptible individual as reference. It is first assumed that, at time t, each individual i

has a true health status represented by $\eta_{i,t}$, which takes value 1 if the individual is infected at *t*, and 0 if the individual is susceptible. Thus, the number of infected individuals at time t is given by:

$$I_t = \sum_{i=1}^{N} \eta_{i,t}. \tag{2}$$

Each individual has one or more clinical signals, which can be summarized by one variable $D_{i,t}$, taking values between 0 and 1. At time *t*, the probability $P_{il,t}$ that a contact between a susceptible individual *i* and an infected individual l results in a new case is a function of the signals of the infected individual only, $D_{l,t}$, as a consequence of the homogeneous susceptibility assumption. It is assumed in particular that this function can be written as a polynomial of degree *M*. That is,

$$P_{il,t} \equiv P_{l,t} = \sum_{j=1}^{M} \varphi_j D_{l,t}^{j} \tag{3}$$

where $0 \le \varphi_j \le 1$ and $\sum_j \varphi_j = 1$, that is, $P_{l,t}$ is a convex combination of $D_{l,t}^{j}$, guaranteeing that $P_{l,t} \in [0,1]$ for all *l,t*. Then the probability that a susceptible individual has, at time *t*, at least one infectious contact defines the stochastic Reed-Frost model as:

$$C_t = 1 - \prod_{l=1}^{N} (1 - P_{l,t})^{\eta_{l,t}}. \tag{4}$$

Note that C_t here can be interpreted as the probability that an individual be infected at time *t+1*, as in the classic Reed-frost model, and it is possible to write $C_t=P\{\eta i,t+1\}=1$.

In some cases $\eta_{i,t+1}$ is unknown, so individuals have to be diagnosed as either infected or susceptible. This consists in a classification procedure which takes into account the clinical signals or, for simplicity, the signals summary $D_{i,t}$, and is defined outside the model, probably by experts. Let $G_{i,t}$=1 indicate that the individual *i* is diagnosed as infected at *t*, and $G_{i,t}$ = 0 indicate that the individual is diagnosed as susceptible. So, the number of individuals diagnosed as infected at time *t* is an estimation of the number of infected individuals at *T*, and is given by:

$$I'_t = \sum_{i=1}^{N} G_{i,t}. \tag{5}$$

In this way, the probability that a contact between a susceptible individual *i* and an infected individual *l* results in a new case is defined by (3) and, in this case, (4) is estimated by:

$$C'_t = 1 - \prod_{l=1}^{N} (1 - P_{l,t})^{G_{l,t}}. \tag{6}$$

Thus, C'_t here is the estimated probability that an individual be infected at time $t+1$.

It is important to highlight that this generalization of Reed-Frost model, taking into account the individual's heterogeneities, has a particular probability structure, which allows some analytical calculus be performed. These calculations supply interesting results from epidemiological point of view.

Menezes *et al.* considered studies involving small groups, within which both homogeneous mixing and homogeneous susceptibility were maintained. The epidemic course depends on the clinical signals involved both in the classification process and to define the probability of an infectious contact. These clinical signals may include symptoms, results of laboratorial and physical exams. It is assumed that no resistance is gained and the individual becomes susceptible again, after being infected. They consider the model in the context of both retrospective, in which patient's health status are observable and modeled as random variables, and prospective studies, in which these true health status are not known. In order to explore the role of the classification process in this epidemic model, we present in this paper a fuzzy approach for the Menezes *et al.* Reed-Frost generalized model, taking into account the *vagueness* of the diagnostic process instead of the *stochastic uncertainty*. We developed a microscopic epidemic model based on the clinical signals and consider a fuzzy relation to evaluate the individual's infectiousness, performing a fuzzy decision process where the infectiousness degree is applied directly in the epidemic dynamic.

2 Fuzzy reed-frost model

Modeling the Reed-Frost dynamic based on the signals scenario is supported by the idea that there is an association between the intensity of the signals present in an infected individual and the probability of an infectious contact, p, with this individual. Thus, it is assumed that, the higher the signal values, the higher the probability that a contact between an infected and a susceptible individual be infectious. The model allows the inclusion of both signals linked to an increased infectiousness and decreased one. Furthermore, distinct signals can affect the probability p with different intensity and ways, affecting also the classification process. So, the calculation of the probability p assumes an important role in this approach, once the individual's information is transmitted to the population dynamic through it.

In this formulation it is assumed that each individual i has a health status, susceptible or infected represented by $G_{i,t}$. The binary variable $G_{i,t}$ takes value 1 if the individual i is infected at t, and 0 if the individual is susceptible. In this way, the number of individuals infected at t, in a group with size N, is given by:

$$I_t = \sum_{i=1}^{N} G_{i,t}. \tag{7}$$

In general, the diagnostic process is based upon the set of signals present in the individual under analysis. This signals set can be summarized by one variable $ID_{i,t}$, taking normalized values into the interval [0,1]. In addition, these clinical signals usually vary its severity depending on both the infectious disease type and the individual variability. Furthermore, for reasons other than infection considered, these signals can be also present in susceptible individuals. In this case, it is expected that its expression should be less intense than in the presence of the infection.

Since the clinical signals expression is different for infected and susceptible individuals, it was assumed two probability distributions, depending on the parameters of either the susceptible or the infected populations. We represent by X_I the signal for any infected individual, and by X_S the signal for any susceptible individual. So, given an individual's health status $G_{i,t}$, X_I and X_S are random variables intrinsically linked to the pathogen; therefore, their distributions remain unaffected by the epidemic course. Once they should take a value within the interval [0,1] we assume the following probability distributions:

$$X_I \sim Beta(\alpha_I, \beta_I)$$
$$X_S \sim Beta(\alpha_S, \beta_S)$$

At time t, the probability $P_{jl,t}$ that a contact between a susceptible individual j and an infected individual l results in a new case is a function of the signals of the infected individual only, $ID_{l,t}$, as a consequence of the susceptibility homogeneity assumption. We assume in particular that this function is:

$$P_{jl,t} \equiv P_{l,t} = \varphi ID_{l,t}{}^{\varpi} \tag{8}$$

where φ and ϖ are parameters of the model and should be chosen in a way to guarantee that $P_{l,t} \in [0,1]$ for all *l,t*. Then the epidemic dynamic in this generalized Reed-Frost model is given by:

$$C_t = 1 - \prod_{l=1}^{N} (1 - P_{l,t})^{G_{l,t}}. \tag{9}$$

C_t here can be interpreted as the probability that an individual be infected at time *t+1*, as in the classic Reed-Frost model, and will be used to generate the health status of the individuals in time *t+1*.

The main difference between the Menezes *et al.* proposal and this fuzzy generalization consists in the structure of summary of signals, which are performed by a random variable in Menezes *et al.* and by a membership degree to be considered infective through a max-min composition in the fuzzy approach.

Consider a set of signals *S* and the matrix representation of a fuzzy relation. Thus, $S_l = [s]_{(1\times k)}$ is the array of *k* signals of the individual *l*, $I = [i]_{(k\times q)}$ is the matrix that associate each signal to the infective statement and $DI_l = [di]_{(1\times q)}$ is the membership degree of the individual *l* in the fuzzy set *Infected*, interpreted here as the degree of infectiousness, found by the fuzzy composition given by:

$$DI = S \circ I \tag{10}$$

whose fuzzy composition ∘ is the max-min composition defined by:

$$DI(di) = \max_{s \in S}[\min(S(s), I(s,i)]. \tag{11}$$

For instance, consider the set of signal is S = [fever, cough], i.e., s_1 is fever and s_2 is cough, and an individual who presents fever degree equal $s_1 = 0.7$ and cough degree equal $s_2 = 0.4$. The matrix I that relates signals and infectiousness is I = [ifever, icough], where i_{fever} is the relationship degree between the symptom fever and infectiousness status and i_{cough} is the relationship degree between the symptom cough and infectiousness status. So, an individual that have a degree of fever, s_{fever}, and a degree of cough, s_{cough}, will belong to the infectiousness fuzzy set with the degree given by:

$$DI = \max\{\min[s_{fever}, i_{fever}]; \min[s_{cough}, i_{cough}]\}. \tag{12}$$

We assume that each individual i has k signals, with levels represented by membership degree in each fuzzy subset of clinical signal (like fever, cough) s_{i1}, s_{i2},...,s_{ik}. So, these levels are numbers between 0 and 1, with $s_{i1} = 0$ indicating that the clinical signal 1 is absent in patient i, and $s_{i1} = 1$ indicating that patient i presents the clinical signal 1 with maximum level (or severity). The infectiousness degree is computed for all individuals and the heterogeneity is considered in the epidemic dynamics through the signal influence on the probability p (the probability of an infective contact between a susceptible and an infected individual) and, consequently, C_t (the risk of a susceptible individual becoming infected). The new individual set of signals in next time is found from C_t.

3 Simulations

In order to analyze the model performance and to compare the fuzzy and stochastic approaches we simulated a virus infection scenario. In addition, the models results are compared with a real data from virus infection.

During the entire 2003 year, all children of a daycare, corresponding to roughly 120 children, with age varying from 1 month to 6 years old, were followed up in São José do Rio Preto, São Paulo, Brazil. The objective of this work, among other things, was to study the circulation of viruses for respiratory infections. All daycare's children with cold symptoms had nasopharyngeal aspirates collected and analyzed with multiplex technique. Therefore, it was possible to determine the true health status of each child. Also, the epidemiological data were collected for all children in the study, independently of the symptomatic status. All children stayed at the daycare during the whole day, what can be considered a quasi-closed group. Although the children are usually distributed in small groups, there are periods along the day that they interact with each other, as in the meals time and in the playful moments. In the same way, there is also interaction among the teachers during the workday. These characteristics, added to the fact that the respiratory infections can be configured as infections of long reach, allow us to consider that the data and the study conditions are in agreement with the model's assumptions.

In order to find the fuzzy relations between signals and infectiousness, four experts in childhood diseases supplied the relational matrices considering the more important clinical signals for infections by viruses. The matrix with the fuzzy relations between signals and infectiousness degree was found by the median of that four experts values. The signs considered for the infections and their respective fuzzy relations values were: fever (0.85), cough (0.85), coryza (0.85); sneezing (0.70) and wheezing (0.60). In this simulations we assumed homogeneous susceptibility and an infected individual was considered immunized to new infections during 3 weeks, which was the minimum period for re-infection observed. So, in the model re-infection is possible, once the protection period is observed.

The model has basically three parameters, which are presented in the equations of the dynamics structure: φ, the polynomial's coefficient; ω, the polynomial's power; and θ, the prior probability of infected status. In addition, the size of the population N was maintained constant since small variations in its value do not affect the result of the model. We assumed N=120, which is around the monthly average of the number of children in daycare.

In order to generate the signals of susceptible and infected individuals we elaborated Beta distributions considering the prevalence of the symptoms of viral infections in the population. The signals prevalence were classified in five categories as follows: *very low*, when the most probable prevalence is roughly 10%; *low*, when this prevalence is roughly 30%; *medium*, when the prevalence is about 50%; *high*, when it is around 75%; and *very high*, when the expected prevalence is around 90%. Figure 1 presents all distributions used and their respective α and β parameters.

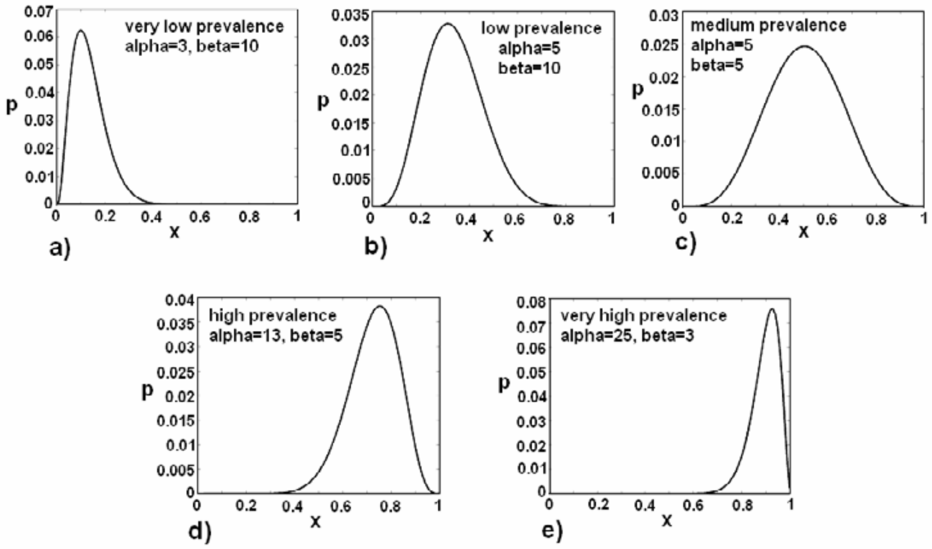


Figure 1: Beta distributions used for the categories prevalence: a) *Very Low*, with α=3 and β=20 parameters; b) *Low*, with α=5 and β=10; c) *Medium*, with α=5 and β=5; d) *High*, with α=13 and β=5; and e) *Very High*, with α=25 and β=3.

As discussed previously, depending on the signal considered, it is possible that an uninfected individual presents signals in some intensity. However, it is not expected that this happen with great frequency in the population. In other words, it is expected that the majority of the susceptible individuals should be not symptomatic. So, it was assumed that all signals of the susceptible individuals have very low prevalence. To determine the prevalence for the signals of infected individuals in the viral infection scenario simulated, it was considered the prevalence observed in the daycare children during the time of the study. So, based on this observation it was assumed the following signals prevalence: fever, sneezing and wheezing are *Very Low*, cough is *High* and corysa is *Very High*. Note that the Beta distributions defined in figure 1 can be used to describe the prevalence of several signals, in different contexts.

Since the simulation of both models involve random process, each simulated condition was repeated 150 times, aiming to find the results through statistical analysis. As expected, the simulations of the models showed that there is a great diversity of dynamical behavior, depending on the parameters values. In some areas of the parameters space the fuzzy and stochastic models are equivalent (for example to small values of φ and ω, with fixed θ). However, there are areas in the phase space where the models present quite different behaviors (see figure 2).

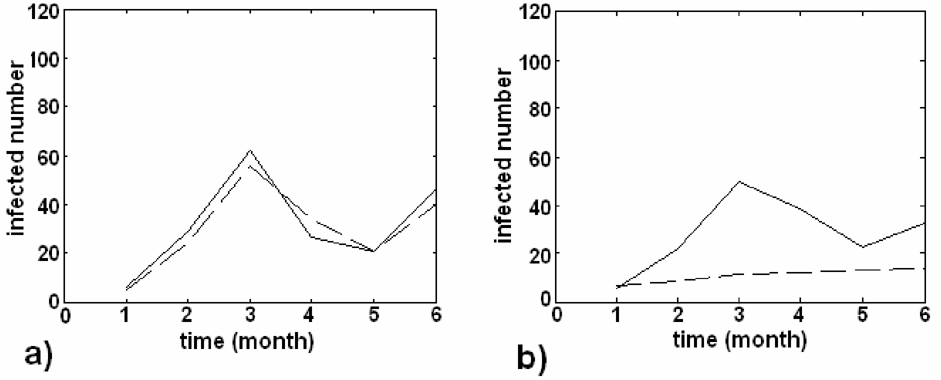


Figure 2: Behavior of the Infected Number in time for the fuzzy model (solid line) and stochastic model (dashed line) for the parameters: a) φ = 0.05 and ω = 0.1, corresponding to the parameter values in which there is equivalence between the models; and b) φ = 0.05 and φ = 1.5, corresponding to the parameter values in which there is no equivalence between the models.

In order to analyze the differences and similarities between the fuzzy and the stochastic models in a more detailed way, a diagram was elaborated varying all parameters of the model and considering the dynamical equilibrium provided by both models. As can be noted in figures below, the diagram presents areas in which the epidemic responses of the models completely agree and areas where they have not similar behavior. In fact, there are no abrupt transition between the regions and frontiers between the regions in this diagram could be considered as fuzzy limitations. However, it is possible to define two crisp states: a so-called *concordant* area, where the systems present very similar behavior in the majority of the points; and a so-called *discordant* area, where the systems present quite different behavior for the majority of the points. The *concordant* area is characterized by the presence of the few types of epidemic response, that is, where the epidemic does not hold or it is endemic for both models. On the other hand, in the *discordant* area there are several concomitant epidemic behaviors (endemic, strong epidemic, etc).

Figure 3 shows that for small values of initial proportion of infected individual (parameter θ ≤ 0.04) there are only three regions in the diagram: 1) a *concordant* area, for small values of φ parameter (φ < 0.01), where the epidemic does not hold for both models; 2) a *discordance* area, where the fuzzy model always present endemic response and the stochastic model presents both no epidemic and endemic responses; and 3) a *concordant* area, where both models present an endemic behavior (this concordance area is maintained for values of φ ≥ 0.07). In none of these regions of the parameters space, it was observed strong epidemics. This is due to the small values of θ parameter, which is responsible for the starting of the infection process in the population.

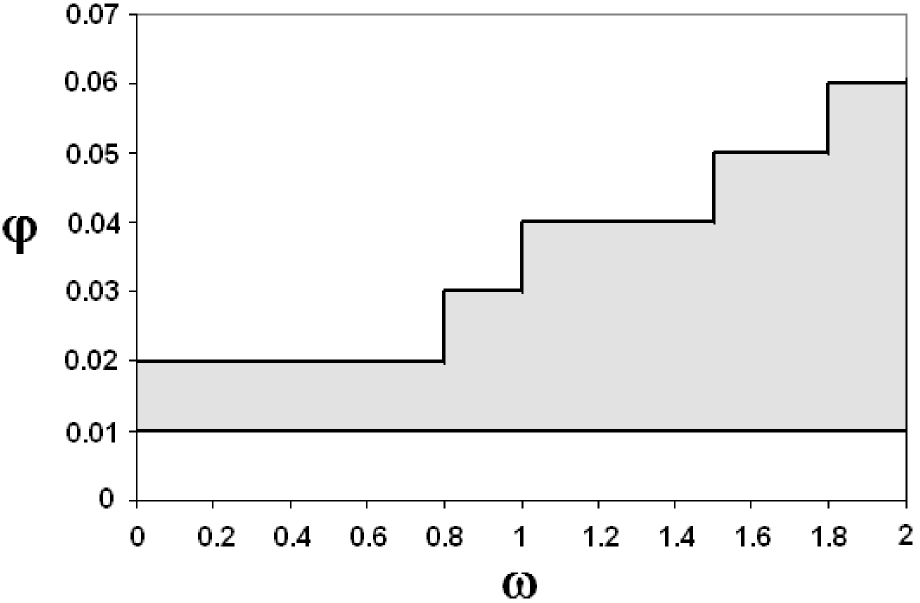


Figure 3: Diagram comparing the epidemic responses of both fuzzy and stochastic models, for θ = 0.02, where two regions are characterized: a concordant region (white area in the graph) and a discordant region (gray area in the graph). For values of φ less than 0.01 the epidemic does not hold for both models. For values of φ greater than 0.06 both models present endemic behavior.

Varying the values of θ, we can note that the regions in the diagram is modified: a fourth region appears in the map, corresponding to a discordant area. Figure 4 shows that, for θ = 0.05 this new discordant region start for φ ≥ 0.8 and small ω values. Moreover, this region increases according to the θ value. This should expected since high levels of θ implies in high virus circulation and, by the models assumptions, the signals are more intense. In addition, due to the properties of the max-min composition of fuzzy relations and the summary of the signals, the epidemic course tends to be stronger in the fuzzy model than in the stochastic approach. This occurs because the differences between the values of possibility and probability of infectious contact is more expressive in this situation. Therefore, in this region both models can results in no epidemic response (particularly for high θ values, because the number of susceptible individuals are very low), weakly, moderate or strong epidemic behaviors, but they do not agree for the majority of the parameters set.

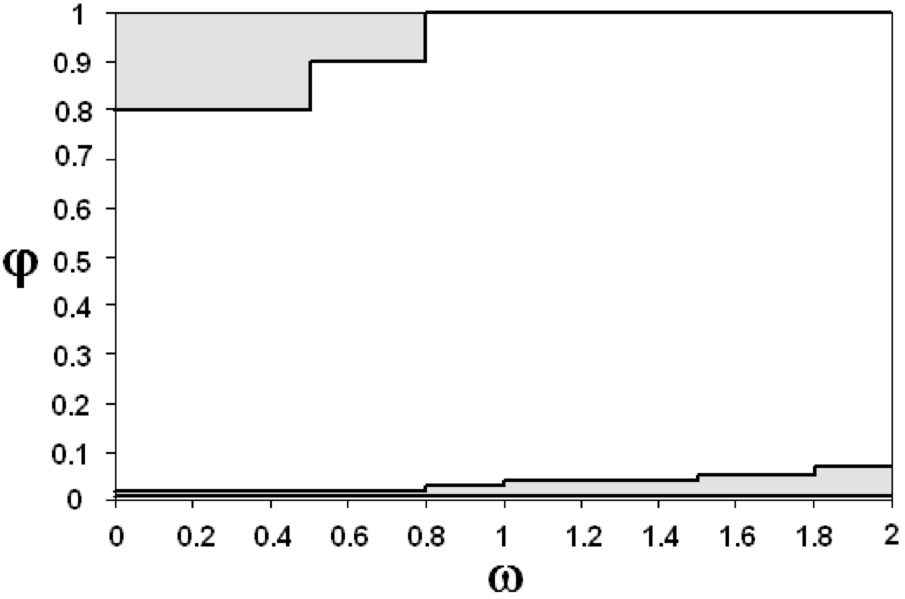


Figure 4: Diagram comparing the epidemic responses of both fuzzy and stochastic models, for $\theta = 0.05$, where four regions are characterized: 1) a *concordant* region (white area in the inferior part of the graph) in which the epidemic does not hold for both models (for values of φ less than 0.01); 2) a *discordant* region (gray area in the inferior part of the graph) in which the fuzzy approach provides endemic behavior and in the stochastic model the epidemic does not hold (for small values of φ); 3) a *concordant* region (white area) in which both models present endemic behavior (for φ values between 0.01 and 0.8); and 4) a *discordant* region (gray region in the superior part of the graph) in which both models provide no endemic, weakly, moderate or strong epidemic behaviors, but they do not agree for a fixed parameters set (φ values greater than 0.8) .

In order to evaluate the models performance when faced with real data, we explored the parameters space seeking to find epidemic behaviors that were comparable to the daycare infections curves. As the number of children varies in time, particularly between the first and the second semesters due to the holidays, the dynamical simulations were performed to a period of half year (each simulation step corresponding to a month). Figure 5 illustrate some examples of these results. We can see in this figure that, for some areas in the parameters space, the models were able to supply a behavior qualitatively similar to the real data. However, the quantitative agreement were not so good when we consider the total number of infected individuals in time.

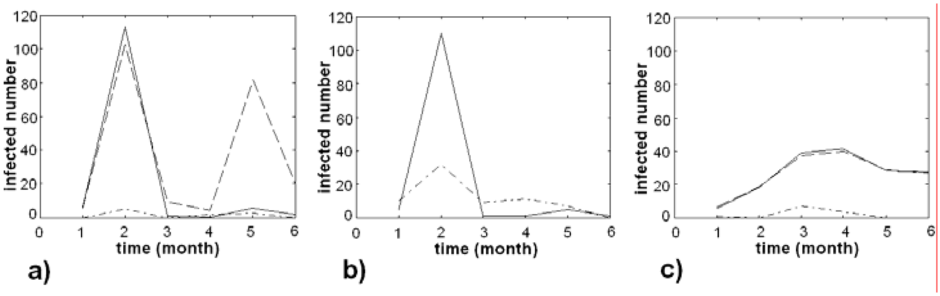


Figure 5: Qualitative comparison between fuzzy (solid line) and stochastic (dashed line) models with real data (dashed dot line), considering the daycare infections in the first semester: a) for RSV infection, in which the results provided by the stochastic model was worse than the one of the fuzzy approach; b) for picornavirus, in which the behavior of the fuzzy and stochastic models are identical; and c) for meta-influenza B infection.

It is possible to note in figure 5a that the models, as well as the real data of the RSV infection in the daycare, present a double peak. In addition, the moment that these peaks occur were the same in the models and in the real data. However, the maximum number of infected is very large in the models when compared with the data. In this case the fuzzy model presents a slightly better results than the stochastic one, since it provided an attenuation of the infection with time (second small peak). In figure 5b we show a comparison between the fuzzy model and the real values of the infection by picornavirus. In this case, the fuzzy and the stochastic results were almost identical. The models supplied a peak of infection in the second month and a second attenuated peak in the fifth month, finishing the epidemic in the 6th month. But in the real data a second peak occurs in the fourth month and it was not so expressive. Figure 5c shows another example considering the influenza B infection, where a qualitative similarity between the models and the real data can be analyzed.

4 Discussion

From the theoretical point of view, both fuzzy and stochastic Reed-Frost models presented here allow several variants. Consider, for instance, the possibility/probability of an infectious contact, which is assumed to be a function of the signals. This function can have any polynomial form, and as such can potentially include any desired function. For instance, by assigning Beta distributions to the individual signals, not only a flexible distribution family, but also one for which all moments are available, with no limitation on the polynomial degree. Besides, it can be generalized to take the possibility/probability of an infectious contact as probabilistic, rather than deterministic, as a function of the signals. Other variants of these models can be obtained by considering more sophisticated classification procedures, which effectively suggests separating the clinical signals effect on different aspects of the epidemic.

Some differences can be pointed between fuzzy and stochastic structures. In the former all signals information related to the possibility function can be performed through fuzzy relational matrix, while in the latter it should be done through the probabilistic function. Clearly, from the interdisciplinary point of view, it is easer to understand the fuzzy relational approach than the mathematical formalism of polynomial functions. In the same way, the heterogeneity of susceptible individuals can be more easily considered in the fuzzy structure. This can be made by simply considering a fuzzy relational matrix that cross information about the immunological characteristics (as information about the child's history, family and personal antecedents, breastfeeding, re-infections etc.) and the degree of susceptibility for the infection. In this sense, the fuzzy relational matrix can supply a fuzzy measure of the individual's protection for certain infection, taking into account the aspects of the identification uncertainties, commonly present in a real epidemic process. In addition, both fuzzy measures for the susceptibility and infectiousness individual's degree, can be elaborated based on the experts opinion.

Considering the simulations results, although the quantitative results of the models are still differing in relation to the real data, the qualitative results are encouraging. It is still necessary to cover the whole space of phases of the parameters and to accomplish small adaptations in the model to improve the fitting of real data of day care. A variable to be investigated is the function of probability. Other non-linear functions of the clinical signals perhaps supply quantitative results more accurate.

The theoretical analysis of differences and similarities between fuzzy and stochastic models provided regions of concordance and discordance, depending on the parameters values. In general, the fuzzy models present stronger epidemic curves than stochastic ones, even in the endemic response. It occurs because the infected degree found through fuzzy relations is always greater than the summary of the signals applied in the stochastic probability function.

Finally, we would like to point out the importance of this work for the area of epidemic modeling, where the shortage of the individual information usually makes unfeasible the elaboration of models that involve the individual aspects (micro) in the epidemic process (macro). Models of this type are rare in epidemiology and their analysis allows a bigger understanding of the factors that may contribute to the force of the infection during an epidemic.

5 Conclusion

Both mathematical models presented were able to provide several epidemic conditions. But, when compared with real data, the fuzzy model provided better results than stochastic model, since it was possible to find a set of parameters in the phase’s diagram that fit better the behavior of these data. However, the fuzzy model should be improved to achieve a better quantitative agreement.

Acknowledgment

This work was supported by the CNPq and FAPESP grants.

References

[1] Pearson DW, “A property of linear fuzzy differential equations”, Appl. Math. Lett. 10 (3), (1997): 99-103.

[2] Seikkala S, “On the Fuzzy Initial Value Problem”, Fuzzy Sets and Systems 24, (1987): 319-330.

[3] Barros LC, Bassanezi RC and Tonelli PA, “Fuzzy modeling in population dynamics”, Ecological Modelling 128 (1), (2000): 27-33.

[4] Barros LC, Leite MBF, Bassanezi RC "The SI epidemiological models with a fuzzy transmission parameter", Computers & Mathematics with Applications 45 (10-11), (2003): 1619-1628.

[5] Jafelice RM, Barros LC, Bassanezi RC, et al., "Fuzzy modeling in symptomatic HIV virus infected population", Bulletin of Mathematical Biology 66 (6), (2004): 1597-1620.

[6] Ortega NRS, Sallum PC and Massad E, "Fuzzy Dynamical Systems in Epidemic Modelling", Kybernetes 29 (1-2), (2000): 201-218.

[7] Ortega NRS, Barros LC and Massad E, "Fuzzy gradual rules in epidemiology", Kybernetes 32 (3-4), (2003): 460-477.

[8] Menezes RX, Ortega NRS and Massad E, "A Reed-Frost Model Taking into Account Uncertainties in the Diagnostic of the Infection", Bulletin of Mathematical Biology 66, (2004): 689-706.

[9] Massad E, Azevedo-Neto RS, Yang HM, Burattini MN and Coutinho FAB, "Modelling age-dependent transmission rates for childhood infections", Journal of Biological Systems 3(3), (1995): 803-812.

[10] Massad E, Burattini MN and Ortega NRS, "Fuzzy Logic and Measles Vaccination: Designing a Control Strategy", International Journal of Epidemiology 28 (1999): 550-557.

[11] Abbey H, "An examination of the Reed-Frost theory of epidemics", Human Biology 24, (1952): 201-2.

[12] Maia JOC, "Some mathematical developments on the epidemic theory formulated by Reed and Frost", Human Biology 24, (1952): 167-200.

[13] Lefévre C and Picard P, "A non-standard family of polynomials and the final size distribution of Reed-Frost epidemic processes", Advances in Applied Probability 22, (1990): 25-48.

[14] Picard P and Lefévre C, "The dimension of Reed-Frost epidemic models with randomized susceptibility levels", Mathematical Biosciences 107, (1991): 225-233.

[15] Scalia-Tomba G, "Asymptotic final size distribution of the multitype Reed-Frost process", Journal of Applied Probability 23 (1986): 563-84.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009
© 2009 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-58603-936-3-194

Representations and Solution Techniques to Loss Reduction in Electric Energy Distribution Systems

Celso CAVELLUCCI, Christiano LYRA, José F. VIZCAINO-GONZÁLEZ and Edilson A. BUENO
School of Electrical and Computer Engineering,
State University of Campinas - UNICAMP
Av. Albert Einstein, 400, Cidade Universitária, 13083-979, Campinas, SP, Brazil

Abstract. Loss reduction in electric energy distribution systems can be considered as a hidden source of energy. These systems operate with a radial configuration. The problem of obtaining a topology with minimum losses can be seen as a generalization of minimum spanning tree problem. The paper discusses representations for the problem of loss reduction through reconfiguration of the network. Models with fixed and variable demands are considered. Solution techniques to find good solutions in adequate computational time to the models proposed in the paper are discussed. A reduced model of a distribution network is used to show important aspects of the problem, under fixed and variable demand representations.

Keywords: loss reduction, network reconfiguration, system distribution, combinatorial optimization

Introduction

The purpose of this paper is to present mathematical models and solution techniques applied to the problem of loss reduction in electric power distribution systems. Structural aspects are discussed and algorithms to solve the problem for real scale networks are briefly presented.

Electric power losses in distribution networks contribute to increase the operational cost of the system, demanding investment anticipation to preserve the service quality. They are inherent to the distribution system; however, there is the possibility to reduce them using adequate methodologies.

In Brazil, electric power distribution losses amount to more than 8%. Innovations in engineering processes can decrease these losses and increase the energy availability in the Country. In order to have a quantitative reference, a reduction of two percent in the distribution energy losses would be able to liberate an energy output quantity equivalent to a 1.500 MW hydroelectric power plant [1].

The approaches discussed in this paper propose technical loss reductions through distribution network reconfigurations. A network reconfiguration can be achieved by changing the open/closed status of switches, keeping the radial topology of the network. The number of possible configurations on a distribution system is associated to the number of switch state combinations, which increases in a factorial relation with the number of open switches existing in the network. Thus, evaluation of all possible configurations is not practical, even with state of the art computers.

The French engineers Merlin and Back [2] where the pioneers to propose, in 1975, the use of network reconfiguration procedures for loss reductions in distribution systems. These authors elaborated two alternative approaches for the technical loss reduction problem. The first of these is an exact approach, applicable just to networks of small size, finds a minimum losses configuration through a branch-and-bound method [3]. The other approach, more efficient with respect to computational performance, uses the optimal power flows distribution in networks with cycles to build progressively a radial configuration solution. This methodology, also known as *sequential switch opening*, was later improved by Shirmohammadi and Hong [4].

Civanlar, Grainger and Lee [5] proposed the well-known *branch-exchange* methodology. The branch-exchange procedure starts with a radial configuration. Losses are reduced by opening and closing switches (branch-exchange), keeping a radial configuration. Many other approaches were proposed to deal with the loss reduction problem [6]. Most of these methodologies consider fixed demands. However, some works already identified the benefits of considering the load variations during a given period [7]-[10]. Although loss reductions is achieved when the configuration of distribution system is changed to adjust to demand variations, it is important to consider that switch operations imply some risk, due to transitory disturbances.

Next section describes a model to represent the primary distribution network. Section 2 presents mathematical formulations for the loss reduction problem, with fixed and variable demands. A case study with a reduced model of a distribution network is used to illustrate the electrical loss behavior under demand variation in Section 3. Solution techniques to deal with the problem of loss reduction are discussed in Section 4. Conclusion follows.

1. Modeling Primary Distribution Network

The main entities to approach the loss reduction problem by network reconfiguration are the following: substations (*SE*), lines (*L*), switches (*SW*) and the consumers represented by load blocks (*LB*). Fig. 1 presents a simplified diagram of an electric power distribution network.

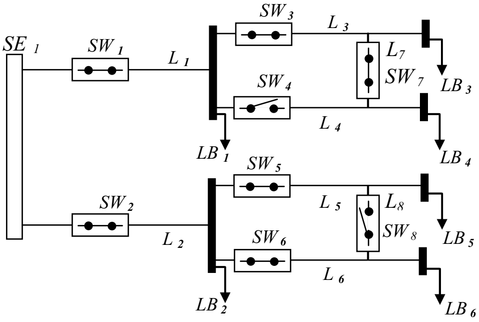


Figure 1. Electric power distribution network

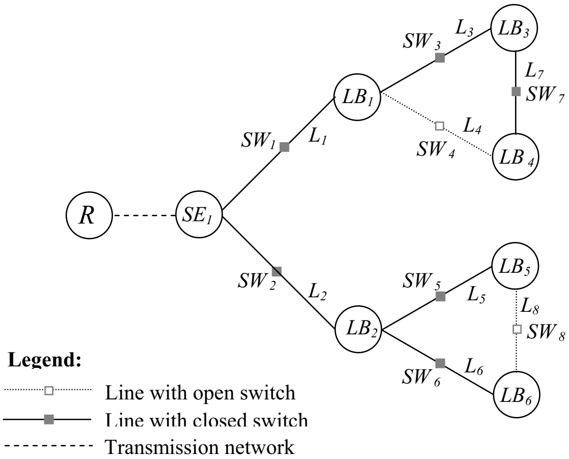


Figure 2. Graph representation of distribution network

A graph structure $G = [\boldsymbol{N}, \boldsymbol{A}]$ can be adopted to model the primary distribution network; where $\boldsymbol{N}$ is the set of nodes and $\boldsymbol{A}$ the set of arcs [11]. Fig. 2 presents a graph representation for the primary distribution network of Fig. 1. The nodes are associated with load blocks or substations (a node root, *R*, is also included to prevent difficulties in dealing with network connectivity). Arcs are associated with either lines or switches - the arcs that connect the substations to the node root represent the transmission lines. The electric power distribution networks usually operate with a radial configuration (note that in Fig. 2 there is a unique path of lines and closed switches from a load block to the substation). Using the graph representation, the radial configuration can be represented by a *tree* $T = [N, A']$, where $A' \subset A$ [11]. Thus, to find the configuration with minimum losses is equivalent to solve a generalization of the minimum spanning tree problem [11].

2. Formulations to Loss Reduction Problem by Network Reconfiguration

Technical losses in the network can be expressed in terms of the active and reactive powers flows in the network arcs [1]. Thus the total losses in the network represented by a graph $G = [\boldsymbol{N}, \boldsymbol{A}]$ can be represented by the function $f(P, Q)$, as follows.

$$f(P,Q) = \sum_{j \in N} \sum_{k \in A_j} r_{jk} \frac{\left(P_{jk}^2 + Q_{jk}^2\right)}{V_j^2} \tag{1}$$

where $\boldsymbol{N}$ is the set of nodes in the distribution network, A_j is the set of arcs with origin at node j, r_{jk} is the resistance of the arc k (line) with origin at node j, P_{jk} is the active power flow in the arc k with origin at node j, Q_{jk} is the reactive power flow in the arc k with origin at node j and V_j is the voltage at node j. Thus a minimum loss configuration can be characterized by $\mathbf{P_1}$,

$$
\begin{aligned}
&\underset{C_v}{Min}\ f(P,Q)=\sum_{j\in N}\ \sum_{k\in A_j} r_{jk}\,\frac{\left(P_{jk}^2+Q_{jk}^2\right)}{V_j^2} && \\
&\text{s.t. } P_{j+1}=\sum_{k\in A_j} P_{jk}-P_{Lj+1} && \text{(a)} \\
&Q_{j+1}=\sum_{k\in A_i} Q_{jk}-Q_{Lj+1} && \text{(b)} \\
&V_{j+1}^2=V_j^2-2\left(r_{jk}P_{jk}+x_{jk}Q_{jk}\right) && \text{(c)} \qquad (2)\\
&\underline{\boldsymbol{P}}\le \boldsymbol{P}\le \overline{\boldsymbol{P}} && \text{(d)} \\
&\underline{\boldsymbol{V}}\le \boldsymbol{V}\le \overline{\boldsymbol{V}} && \text{(e)} \\
&T=\left[\boldsymbol{N},\boldsymbol{A}'\right] && \text{(f)}
\end{aligned}
$$

where $\boldsymbol{C_v}$ is the set of radial configurations for the network, PL_j is the active load on node j, QL_j is the reactive load on node j, $\boldsymbol{P}$ is the vector of active power flows, $\underline{\boldsymbol{P}}$ e $\overline{\boldsymbol{P}}$ are vectors of bounds for active power flows through switches, $\boldsymbol{V}$ is the vector of the node voltages, $\underline{\boldsymbol{V}}$ and $\overline{\boldsymbol{V}}$ are the bounds for node voltages.

Two aspects can simplify the problem $\mathbf{P_1}$. The voltage magnitudes in distribution system can usually be approximated by one per unit (V_j = 1 p.u.) [1]. In this case the voltage magnitude constraints can be dropped. In addition, in well- compensated network, the reactive power can be considered approximately proportional to active power (e. g., $Q_j = \alpha\, P_j$). Under this assumption, both power flow equations, (1.a) and (1.b), are equivalent, and the equation (1) can be expressed as follows [1].

$$f(P)=(1+\alpha^2)\sum_{j\in N}\ \sum_{k\in A_j} r_{jk}P_{jk}^2 \qquad (3)$$

Note in the Eq. (3) that the term $1+\alpha^2$ not modified the optimization result; it only needs to be considered to compute losses. Thus, the problem $\mathbf{P_1}$ can be simplified into $\mathbf{P_2}$.

$$
\begin{aligned}
&\underset{C_v}{Min}\ f(P,Q)=\sum_{j\in N}\ \sum_{k\in A_j} r_{jk}\,\frac{\left(P_{jk}^2+Q_{jk}^2\right)}{V_j^2} && \\
&\text{s.t. } P_{j+1}=\sum_{k\in A_j} P_{jk}-P_{Lj+1} && \text{(a)} \\
&\underline{\boldsymbol{P}}\le \boldsymbol{P}\le \overline{\boldsymbol{P}} && \text{(b)} \qquad (4)\\
&T=\left[\boldsymbol{N},\boldsymbol{A}'\right] && \text{(c)}
\end{aligned}
$$

This formulation allows finding the switches that should be opened and closed in the optimal network configuration. However, in practice the electric energy demands in distribution networks are time varying, demanding appropriate models.

2.1 Formulations with Uniform Demand Variations

As mentioned previously, the demands of electric energy in distribution systems vary with the time. Initially, we consider a uniform variation of the demands. In other words, we adopt the assumption that all loads change with the same pattern. Fig. 3 illustrates a load curve with uniform demand variation, where λ_i is a multiplier associated to the demand level of the time interval i and Δ_i is the duration of i.

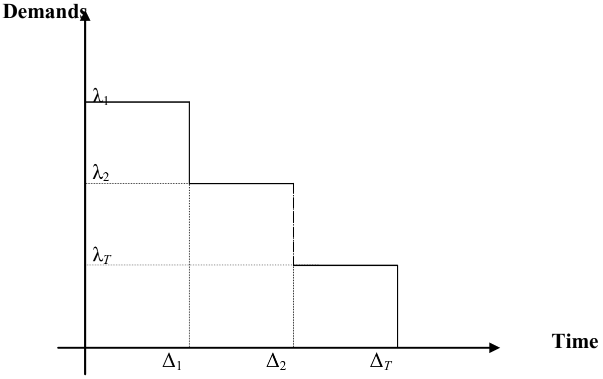


Figure 3. Load Curve with Uniform Demand Variations

The energy losses considering uniform demand variation can be defined by the function $f(P, T)$,

$$f(P,T)=\sum_{i=1}^{T}\Delta_i\left(\sum_{j\in N}\sum_{k\in A_j} r_{jk}\left(\lambda_i P_{jk}\right)^2\right) \tag{5}$$

where T is the number of time intervals represented in the load curve.

In this case it is easy to show that energy losses are proportional to active power losses; so it is equivalent to solve the losses reduction problem for fixed demands and then multiply the result by the constant $K=\left(\sum_{i=1}^{T}\Delta_i\lambda_i^2\right)$ [1]. In practice, the assumption of uniform load variations is used mainly for planning studies. For operation purposes it is more adequate to build a model with non-uniform demand variations.

2.2 Formulation for Non-Uniform Demand Variation

Formulations with non-uniform demand variation assumption can be building under two scenarios: the first one, allow configuration changes after each significant change in the demand profile, the other one, proposed in [1], maintain the same configuration for whole planning period. The problem P3 considers the first scenario.

$$
\begin{aligned}
&\underset{C_v}{Min}\sum_{i=1}^{T}\sum_{j\in N}\sum_{k\in A_j}\Delta_i\left(r_{jk}P_{ijk}^2\right) && \\
&\text{s.t. } \boldsymbol{A}\boldsymbol{C}_i\,\boldsymbol{P}_i=\boldsymbol{b}_i && \text{(a)} \\
&\underline{\boldsymbol{P}_i}\le\boldsymbol{P}_i\le\overline{\boldsymbol{P}_i} && \text{(b)} \\
&T_i=\left[\boldsymbol{N},\boldsymbol{A}_i'\right] && \text{(c)}
\end{aligned}
\qquad (6)
$$

where T is the number of time intervals considered, $\boldsymbol{A}$ is the incidence node-arc matrix associated to the graph $\boldsymbol{G}$, $\boldsymbol{C_i}$ is the diagonal matrix representing switch states in the network during time interval i, $\boldsymbol{P_i}$ is the vector of active power flows in the interval i, $\boldsymbol{b}_i$ is the vector of demands and $\boldsymbol{A}_i'$ is the set of arcs that represent closed switches in time interval i.

Note that to solve problem $\mathbf{P_3}$ is equivalent to solve several uncoupled problems, one for each interval i.

The second scenario, problem $\mathbf{P_4}$, restricts the number of switching through the imposition of one fixed configuration for the whole planning period. This restriction can be obtained replacing the equation (5.a) by the equation that follows.

$$\boldsymbol{A}\boldsymbol{C}\boldsymbol{P}_i=\boldsymbol{b}_i \qquad (7)$$

where $\boldsymbol{C}$ is the diagonal matrix representing the switch states in the network for the whole planning horizon. This restriction leads to a problem with dimension and complexity degree greater than the problem $\mathbf{P_3}$, because it cannot be decomposed into subproblems.

3. Case Study

A reduced model diagram of the distribution network with 9 nodes and 11 arcs is showed in Fig. 4.

Table I shows the energy demand in each load block d_i (in Fig. 4) for two levels of load values: low and high.

TABLE I. Demands in each Consumption Block

	d_1	d_2	d_3	d_4	d_5	d_6
Low (12 hours)	3,0	2,5	2,5	1,5	1,5	1,0
High (12 hours)	22,0	3,0	4,0	2,5	2,5	2,0

Fig. 5 shows the CONFIG A and CONFIG B, optimal radial configurations for low and high load levels, respectively, considering the resistance for all arcs equal to 1 ohm.

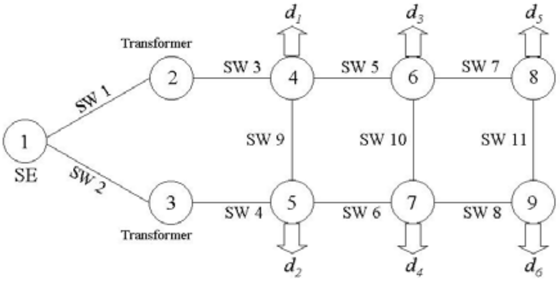


Figure 4. Model reduced of the distribution network

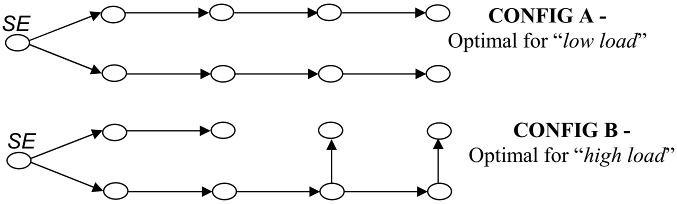


Figure 5. Optimal Configurations for Low and High Loads

The losses in power (kW) and energy (kWh) for the two load profiles are present in Table II. Last column shows energy losses during a one-day period.

TABLE II. Total Loss Energy Values

	Losses				
	Low Load		*High Load*		**Total**
	Power (kW)	Energy (kWh)	Power (kW)	Energy (kWh)	Energy (kWh)
Duration	12 hours		12 hours		24 hours
Config A	*99,5*	*1194,0*	*941,3*	*11295,6*	*12489,6*
Config B	*147,0*	*1764,0*	*843,5*	*10122,0*	*11886,0*

Another radial configuration is showed in Fig. 6. Its power and energy losses for low and high load are presented in Table III.

TABLE III. Energy Losses Values in Alternative Configuration

	Losses				
	Low Load		*High Load*		**Total**
	Power (kW)	Energy (kWh)	Power (kW)	Energy (kWh)	Energy (kWh)
Duration	12 hours		12 hours		24 hours
Config C	*103,3*	*1239,9*	*867,5*	*10410,0*	*11649,6*

Note in Table III that the CONFIG C is not the best alternative for either the low and high loads. However, it is better than CONFIG A and CONFIG B under the restriction of operation with a fixed configuration for the whole day period.

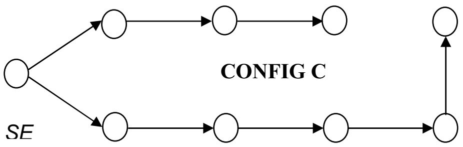


Figure 6. An alternative configuration

4. Some Techniques to Approach the Problem

The loss reduction problem through network reconfiguration is a combinatorial optimization problem with constraints; the optimal solution could theoretically be achieved by examining all feasible spanning trees. However, such exhaustive search is computationally intractable, especially in the case of time-varying demand. Alternative methods to find good solutions in adequate computational time were proposed using heuristic approaches, evolutionary algorithms and artificial intelligence techniques.

As previously mentioned, Merlin and Back [2] elaborated the *sequential switch opening* method in a two-step procedure. In the first step, all available switches are closed and an optimal flow pattern for meshed network is found. The second step inspects the solution found in first step to see if it has meshes; if so, it uses a *greedy* algorithm to achieve a radial configuration by opening the switch in a mesh with the smallest flow.

The *branch-exchange* approach proposed by Civanlar et al. [5] starts with a feasible solution (radial configuration). Through a local search (branch exchanges), the network configuration is modified successively trying to reduce the power losses. Essentially, it relies on two points:

1. Opportunities to decrease losses exist when there are significant voltages differences between nodes connected by switches.
2. Loss reduction is achieved by transferring loads from sides with lower voltage to sides with higher voltages, by closing and opening switch operations.

Cavellucci and Lyra [12] proposed an informed search [13]. The proposed two-step procedure investigates a radial configuration of low losses over a state space, where each state (or node) is characterized by a network configuration with optimal flow pattern. The first step is similar to the *sequential switch opening* method [4], solving a problem without the radiality constraint. A *heuristic backtrack* procedure [13] is proposed in the second step. Searches begin at the *start node*, identified with initial optimal solution of relaxed problem (first step solution). Opening switches and solving the subsequent problem generates successor nodes. A reference feasible solution was adopted to allow *pruning by dominance* [13], reducing the state space to explore. Additional problem knowledge is included by a heuristic evaluation function to estimate the increase in losses from a mesh configuration to a radial configuration.

As the heuristic backtracking procedure explores a node, it considers all possibilities of opening switches. Most of them will lead to paths that will be pruned latter. A *selective heuristic backtracking* are proposed by Lyra, Fernandes and Cavellucci [14] that considers only a "promising set" of switches as *branching factor* in the search procedure. The "promising set" comprises the *p* switches with the smallest flows. When *p* is set to one, the *selective heuristic backtracking* mimics the *sequential switch opening* method; when p is large enough to consider all possibilities of opening switches at a node in the search space, it coincides with the *heuristic backtrack.*

Vargas, Lyra-Filho and Von Zuben [10] applied Classifier Systems [15] to find a network configuration with low losses for each significant demand variation. Classifier Systems communicate with the environment through detectors, effectors and feedback. Basically, these systems refer to a method for creating and updating a kind of "*if antecedent-then consequent*" rules (the classifiers), which code alternative actions to deal with present state of the environment. The detectors are responsible to the reception and codification of the messages received by the system and the effectors responsible to act on the environment. A schema appropriate to reward the classifier better adapted to the environment determines the actions to be applied (environment's feedback).

Two approaches proposed by Bueno, Lyra and Cavellucci [16] can solve the loss reduction problem for non-uniform demand variation, with fixed configuration constraint. An extension of the *sequential switch opening* method was developed using the energy flow of each arc of the network to guide the search. Another approach uses a GRASP (*Greedy Randomized Adaptive Search Procedure*) metaheuristic [17]; this approach considers an *approximate tree* algorithm [1], based on the minimal spanning tree *Kruskal* algorithm [11].

Queiroz and Lyra proposed a *hybrid genetic algorithm* approach to solve the loss reduction problem under variable demands [18]. Radial solutions are represented by strings of real-value weights (one for each branch), initially randomly generated. A Kruskal algorithm [11] is used to find the spanning tree associate to each solution. The algorithm includes a local search based on generalization of the *branch-exchange* procedure [5]; the generalization was developed to consider load variations.

5. Conclusion

This paper presented ideas for mathematical formulations and solution techniques to the problem of loss reduction by network reconfiguration in power distribution systems. The model proposed to deal with the problem considers two scenarios of fixed and variable demands.

Even though the best loss reductions are achieved when a minimum loss configuration is obtained after each significant load variation, this approach can lead to operational problems that can affect the service quality. An alternative approach that considers load variations but keeps the configuration unchanged for the planning period is also discussed.

A case study with a simple illustrative network shows the benefits of considering demand variations but adopting unchanged configuration for the planning period.

Acknowledgements

The authors acknowledge the kind support from CNPq (Brazilian National Research Council for Scientific and Technological Development) and CPFL (Light and Power Company of São Paulo).

References

[1] E. A. Bueno, *Redução de Perdas Técnicas através de Reconfigurações de Redes de Distribuição de Energia Elétrica sob Demandas Variáveis* (in Portuguese), 2005.

[2] A. Merlin and H. Back, Search for a Minimal-Loss Operating Spanning Tree Configuration in an Urban Power Distribution System, *Proc. 5th Power System Computation Conference (PSCC)*, Cambridge (UK), **article 1.2/6**, 1975.

[3] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, USA, 1988.

[4] Q. Zhou, D. Shirmohammadi and W. H. E. Liu, Distribution Feeder Reconfiguration for Operation Cost Reduction, *IEEE. Transactions on Power Systems*, vol.12, pp. 730-735, 1997.

[5] S. Civanlar, J. J. Grainger, H. Yin and S.S.H. Lee, Distribution Feeder Reconfiguration for Loss Reduction, *IEEE Transactions on Power Delivery*, vol. 3, No. 3, 1988, pp. 1217-1223.

[6] C. Lyra Filho, C. Pissarra and C, Cavellucci, Redução de Perdas na Distribuição de Energia Elétrica, *Anais do XIII Congresso Brasileiro de Automática – CBA*, 2000.

[7] C. S. Chen and M. Y. Cho, Energy Loss Reduction by Critical Switches, *IEEE Transactions on Power Delivery*, vol. 8, pp. 1246-1253, 1993.

[8] R. Taleski and D. Rajičić, Distribution Network Reconfiguration for Energy Loss Reduction, *IEEE Transactions on Power System*, vol. 12, pp. 398 – 406, 1997.

[9] K. Y. Huang and H. C. Chin, Distribution Feeder Energy Conservation by using Heuristics Fuzzy Approach, *Electrical Power and Energy Systems*, vol. 24, pp. 439-445, 2002.

[10] P. A. Vargas, C. Lyra, and F. J. Von Zuben, Learning Classifiers on Guard Against Losses in Distribution Networks, *IEEE/PES T&D 2002 Latin America*, 2002.

[11] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications,* Prentice Hall, Englewood Cliffs, NJ. 1993.

[12] C. Cavellucci and C. Lyra, Minimization of energy losses in electric power distribution systems by intelligent search strategies, *International Transactions in Operational Research*, vol 4, no. 1, pp. 23-33, 1997.

[13] J. Pearl, *Heuristic: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984.

[14] C. Lyra, C. P. Fernandes and C. Cavellucci (2003), Informed Searches Uncover Paths to Enhance Energy Output in Power Distribution *Networks", Proceedings of the "12th Intelligent Systems Application to Power Systems Conference"* (CD ROM), paper ISAP03-102, Greece, 2003.

[15] L. B. Booker, D. E. Goldberg and J. H. Holland, Classifier Systems and Genetic Algoritms, *Artificial Intelligence*, vol. 40, pp 235-282.

[16] E. A. Bueno, C. Lyra, and C. Cavellucci, Distribution Networks Reconfiguration for Loss Reduction with Variable Demands, *IEEE/PES T&D 2002 Latin America*, 2004.

[17] M. G. C Resende. and C. C.Ribeiro, *Greedy Randomized Adaptive Search Procedures,* AT&T Labs Research Technical Report TD-53RSJY, version 2, 2002.

[18] L. M. O. Queiroz and C. Lyra, A Genetic Approuch for Loss Reduction in Power Distribution Systems under Variable Demands, *IEEE World Congress on Computational Intelligence - WCCI2006*, Vancouver, Canada, 2006.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-205

Intelligent Vehicle Survey and Applications

Luiz Lenarth Gabriel VERMAAS, Leonardo de Melo HONÓRIO, Edison Oliveira de JESUS, Muriell FREIRE and Daniele A. BARBOSA
Institute of Technologies, Federal University of Itajubá
Av. BPS 1303 – Itajubá – 37500-903 – MG – Brazil

Abstract: Intelligent or Autonomous Vehicles are not visionary technologies that may be present in a far away future. From high tech military unmanned systems and automatic reverse parallel parking to cruise control and Antilock Brake Systems (ABS), these technologies are becoming embedded in people's daily life. The methodologies applied in these applications vary widely across areas, ranging from new hardware development to complex software algorithms. The purpose of this work is to present a survey about the benefits and applications of autonomous ground vehicles and to propose a new learning methodology, which provides the vehicle the capacity of learning maneuvers with a human driver. Four main areas will be discussed: system's topology, instrumentation, high level control, and computational vision. Section 1 will present the most common architecture of autonomous vehicles. The instrumentation used in intelligent vehicles such as differential global positioning system (DGPS), inertial navigation system (INS), radar, ladar and infrared sensor will be described in section 2, as well as the techniques used for simultaneous registration and fusion of multiple sensors. Section 3 presents an overview of some techniques and methods used for visual control in autonomous ground vehicles. Section 4 will describe the most efficient techniques for autonomous driving and parking, collision avoidance and cooperative driving. Finally, section 5 will propose a new algorithm based on Artificial Immune Systems, where a fuzzy system for autonomous maneuvering will be learnt by a data set of actions taken by a human driver. In order to validate the proposed method, the results of its application in an automatic parallel parking maneuver will be showed.

Keywords: autonomous vehicles, intelligent transportation systems, vehicular instrumentation, computational vision, co-evolutionary artificial immune systems, fuzzy system learning.

Introduction

The most common definition about intelligent vehicles is that they are capable of making driving decisions without human intervention. These decisions have different levels of intelligence, and may be present in the entire vehicle, making it autonomous, or in independent systems. Many of these independent systems are present in a car today. Cruise Control, Antilock Brake Systems (ABS), Electronic Stability Program (ESP) and Anti-Skid Control" (ASC) are well known examples of low intelligence levels which can be found in several models of vehicles. A higher level of autonomy is already present in a few models, such as Advanced Cruise Control, Lane Keeping Support, Lane Changing Warning, Collision Warning, Obstacle Avoidance, Route Control and Reverse Parallel Parking. For each of the described applications, it is necessary that the vehicle senses itself (global position, speed, acceleration, inertial movement, etc) and the environment (lane markings, local map, obstacles, other vehicles, traffic signs or lights, etc). With different sensors responsible for that data acquisition, the final information may present a high level of uncertainty. This scenario makes it difficult to have a demanding level of liability. Therefore, innovations towards a fully autonomous car happen very slowly. However, in controlled environments, like large industrial complexes or farms, autonomous vehicles are already a reality. In those situations, unmanned vehicles operate on a pre-defined path using highly precise DGPS systems along with others sensors. Many reports about applications on both scenarios can be found in the literature [1 - 6].

The growing research interest in those fields can be exemplified by the IEEE, IEE and other relevant societies' workshops, conferences, publications and challenges. The most significant improvements are listed next:

High Precision Instrumentation – Nowadays, instrumentation hardware have accomplished incredible level of precision. DGPS equipments can ensure centimeter-level of precision and, therefore, be employed to detect the vehicle position, increasing the amount of information provided by the sensors. Radar and Ladar sensors provide a long-range capability offering valuable data for identifying the scene.

Artificial Intelligence Techniques - Powerful algorithms are capable of dealing with imprecise and incomplete data, enabling the modeling of the environment in a very reliable manner. By using these algorithms, it is possible to: accept data from sensors, assembling an understandable view of the current situation; plan certain decisions, in order to accomplish a given task; reorder the plan, if unexpected situations occur; and foresee conditions, avoiding hazard situations.

Computational Vision – it is one of the most investigated areas in the field of intelligent vehicles. It mixes high-performance hardware with real-time software. Researches in this field offer a wide spectrum of features: stereo vision, motion, appearance, edges, and others.

This work overviews these issues, presenting the main paradigms of autonomous vehicles and proposing a learning methodology, which provides the vehicle the capacity of learning maneuvers with a human driver. The survey section is divided into four parts: architecture of intelligent vehicles, instrumentation, computational vision and high level control. The proposed methodology is organized as follows: a general description of the main method, details about the co-evolutionary gradient based artificial immune system, and the algorithm application results.

1. Architecture of Intelligent Vehicles

Solutions involving intelligent vehicles need to deal with a rapid growth of sensor information, and provide a highly accurate performance (as much in quality as in time) in a variety of situations. To perform these tasks, the architecture of intelligent vehicles varies from 2 up to 4 layers, as it can be seen in Figure 1. Each layer is described next.

a) Sensor Network – it is a set of sensors, such as cameras, ladar, radar, velocity, position, IMU, etc, responsible for capturing data from the environment. Today, a single sensor is not liable enough; therefore, multi-sensor systems employ fusion strategies to enhance the degree of safety. The main problem with this approach is to transform incoming data, provided by sets of sensors with different levels, into reliable information.

b) Environment Identification – it is responsible for achieving the understanding of a scene by building knowledge from a sensor network. This awareness is built from a consistent mix of data passed by sets of sensors. It must provide information about obstacles, possible collisions, other vehicles, pedestrians, traffic lights and signs, lanes, etc. The vehicle can attain this identification using its own systems or it can exchange information with others, consequently expanding its knowledge about the environment. This layer represents the human capability to recognize elements of a given scenario. Although the scene is correctly identified, it does not ensure the knowledge of what it means. For instance, an intelligent vehicle can recognize the approach of another car; however, it may not be able to identify a possible collision situation.

c) Situation and Behavior Control – The description of the system provided by the Environment Identification does not have sufficient understanding to achieve autonomy. It is necessary to transform this information into knowledge, by recognizing what situation the vehicle is involved in, and set the proper operation state. Traditional control systems express problems in terms of numerical functions. However, for this application domain, the control laws go beyond that; it must represent the human capability of identifying and solving a problem. For that purpose, this layer must provide a repertoire of tactics and action models (collision avoidance, lane keeping, parking maneuvers, navigation settings, and others), intelligent control algorithms (fuzzy logic, neural network, IA Planning, etc), and a finite machine state responsible for defining the change of tactics and actions. In the former example, after the correct identification of a collision situation, the system sets a behavior for the vehicle.

d) Low Level Control – This layer encapsulates the kinematical model of all of the systems of the vehicle. It may present both traditional and intelligent controllers, focusing on controlling the internal quantities of the vehicle, i.e., velocity of the wheels, acceleration, yaw rate, brake rate and intensity, etc.

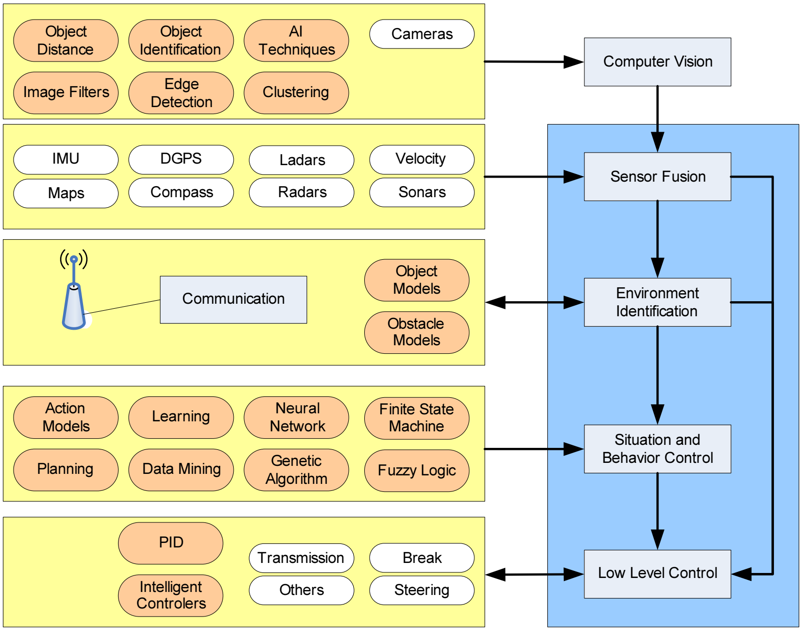


Figure 1. Typical Architecture of Intelligent Vehicles

Note that, for applications involving lower level of intelligence, only two layers are needed. For instance, on an ABS system, the sensor is connected to the low-level control, acting directly on breaks and dismissing the other functionalities. As the intelligence grows, the number of layers tends to grow as well. Solutions involving fully automated vehicles [7] present all layers.

2. Vehicular Instrumentation

An important part of all of the autonomous systems is the perception and acquisition of knowledge about the environment where the system is. The primary element regarding this perception is the sensor. Basically, the perception function in intelligent vehicles applications consists in substituting the human driver vision for electronic sensors. Today, there are a wide variety of sensors available for these applications.

The main sources of information for the perception system in intelligent vehicles are odometric, inertial, vision and laser sensors, DGPSs and radars. Inertial sensors supplying angular rates and linear accelerations provide information about the current interaction of the vehicle with its environment. Vision, in contrast, supplies information about the future environment the vehicle will meet.

According to the literature the first intelligent vehicle with fully automatic driving was built in Japan in 1977. The pioneering project of the team coordinated by Professor Sadayuki Tsugawa [1] used artificial vision, an engine coupled to a steering column, and a proportional control system. In spite of the simplicity of the primitive design it showed the technical feasibility of autonomous vehicles.

Although they still have not become a commercial reality, autonomous vehicles that can drive without human intervention are already a technological reality. Some research groups throughout the world have developed automatic control systems in actual vehicles such as the Autopia Program [39], the Partners for Advanced Transit and Highways - PATH [40], the DARPA’s Navlab [26], the Martin Marietta`s Alvim [41], Lanelok project [42], the advanced platform for visual autonomous road vehicle guidance – VAMP [43], ARGO vehicle [44], Praxitele [45] and Vehicle with Intelligent Systems for Transport Automation – VISTA [46], among others.

2.1. Sensors, Applications and Characteristics

Designing a vehicle that is able to navigate in roadways, urban areas or any external environment without human intervention is considerably different from designing a vehicle to navigate in internal, relatively controlled, environments.

In order to perceive the real world, the vehicle needs to be equipped with several types of sensors that can be functionally grouped in the following way:

a) Route recognition sensors: in general they are electromagnetic sensors, lasers, transponders or computational vision to identify marks in the road;

b) Obstacle recognition sensors: they usually are laser scanners, ultrasonic sensors, radars, infrared sensors or stereoscopic visions that allow the identification and avoidance of crashes with other vehicles or accidents with pedestrians or other obstacles;

c) Navigation sensors – they are DGPSs, accelerometers, gyroscopes, compasses, speed sensors, or odometers used for assuring that the vehicle will run in a safe way and controlled by the defined route.

In order to be applied to vehicles, the instrumentation must also meet the following requirements: low cost, small size, robustness, effective under most weather conditions, real time detection and integration with other modules.

A survey on the types of sensors used in intelligent vehicles showed that there are simple and low cost designs and designs such as Stanley [47], the autonomous vehicle that won the 2005 DARPA Grand Challenge. This vehicle used five laser sensors to measure the distance, two radars, a stereo camera and a monocular camera. In order to infer its position, the vehicle also used DGPS receptors, INS, a digital compass, besides attaining the speed and the traveled distance through the Controller Area Network – CAN bus of the used automobile.

Table 1. Sensors, applications and characteristics

Sensor	***Used for***	Important Characteristics
DGPS	Navigation	Global coverage and absolute position; Provide accurate position resolution: ~ 1cm; Usually the data output rate is low: ~ 10 Hz; Subject to outages; In urban areas, signals are often blocked by highrise buildings
INS	Navigation	Relative position; Data rate can reach up to more than 100 Hz; Self-contained and is immune to external disturbance;
Camera	Road and Object Detection	Do not perform well in extreme weather or off-road conditions;
Radar	Object Detection	Allow to measure the distance, velocity, and heading angle of preceding vehicles; Detection range: ~ 100 m; Update rate: ~ 10Hz
Laser Radar	Road and Object Detection	Useful in rural areas for helping to resolve road boundaries; Fail on multilane roads without the aid of vision data;
Digital Compass	Navigation	Can provide heading, pitch and roll with a sampling rate of around 20 Hz***;*** Pitch and roll output is polluted with noise; The speed of response is very low when the vehicle was running in the uneven off-road environment where the attitudes change rapidly.

Ozguner et al [7] propose a vision system with different sensors able to perceive an environment of 360º around the car. Eight ultrasonic sensors with a range of 4m each are installed in the side and in the rear parts of the car. In the front part, a radar with a range of up to 100m, a pair of stereo cameras that allow the vision within a 35 meter radius and three 180º scanning laser rangefinders with a range of up to 50m were installed.

Hatipoglo et al [48] use a vehicle with steer by wire – SBW, drive by wire – DBW, throttle and brake control capabilities, as well as access to a number of vehicle state measurements. The car was equipped with a low cost CCD monochromatic video camera installed in the position of the rear-view mirror, RF radar and a scanning laser rangefinder installed in the front bumper.

In the Multifocal Active/Reactive Vehicle Eye – MARVEYE project [49] an Expectation-based Multifocal Saccadic Vision (EMS-Vision) system is used. Contrary to the most common implementations, which work with statistical and optimized configurations aiming at specific tasks or domains, the EMS-Vision is a flexible system able to configure itself dynamically during the operation, similarly to the human eyes that adjust themselves to the distance, for example. Whereas this vision system is used to perceive the future environment that the vehicle will find, INS sensors provide current information on the interaction of the vehicle and the environment.

The Spanish program AUTOPIA [59] has equipped three commercial cars aiming at their automatic driving at a closed circuit. The track, 1 km long, allows the reproduction of most traffic situations such as turning, stopping and going, overtaking, driving in platoon, parking and etc. Besides the GPS installed in the three cars, the first one also uses a laser scanner in order to detect obstacles, the second uses an artificial vision system to identify the circuit and the third car uses a stereo vision system to detect pedestrians and obstacles on the track. A special resource used by this project is a wireless local area network (W-LAN) in order to transmit the GPS differential correction information to the receiving GPS installed in the cars. This communication system replaced the traditional point-to-point radio modem.

An intelligent vehicle project, whose main objective is to develop a safety-warning and driver-assistance system and an automatic pilot for rural and urban traffic environments, is presented in [20]. The vehicle called Springrobot uses computational vision, IR sensors, radar, lidar, DGPS, accelerometer, and gyroscope to perceive the traffic conditions, make decisions and plan the route to travel on structured highways and unpaved road of rural or urban areas.

2.2. Multi-sensor and Fusion

Section 2.1 of this work approached the instrumentation that is commonly used in intelligent vehicles projects. It is widely known that only type of sensor is not able to provide all the necessary information for this sort of application. Each sensor has its own characteristics, advantages and disadvantages. The DGPS, for example, can inform the position with an accuracy of 1 cm, but at some places the satellite signal might be blocked or have its accuracy reduced. Another widely used sensor is the INS, whose advantage is having low sensitivity to high frequency noises and external adversities. However, unless the INS is calibrated online its measurements errors accumulate. Several fusion sensor techniques have been researched and applied aiming at the integration and a better use of the complementarity among the sensors that have different features such as measuring scale, sensitivity, and reliability. The fusion of the information of different sensors also allows the system to continue working in case there is a failure or loss of one of the sensors: redundancy.

An algorithm based on the unscented Kalman filter – UFK is proposed in [50] to register and simultaneously fuse a GPS, an INS, a radar and a camera for a cooperative driving system. Figure 2 illustrates a platoon of vehicles on a two-lane highway and the block diagram of the instrumentation and of the sensor fusion system. First, a temporal-spatial logging model was developed to record the system bias and the measurements of the different sensors carried out at different times. After the registration, the fusion of the sensors information is carried out by an UFK.

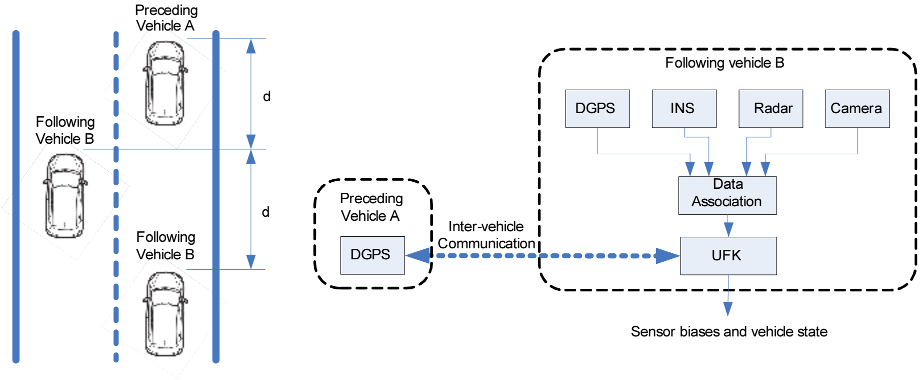


Figure 2. Platoon and sensor fusion block diagram

The GPS/INS integration has been widely researched [51 – 58] over the last decade. This implementation is essentially based on techniques that use the Kalman filter or its variants. Generally, the extended Kalman filter is used in the architecture of GPS/INS integration. However, if any unknown parameter appears in the system model or if the model changes with the environment, as in the case of Intelligent Transportation Systems, the estimated accuracy is degraded. An algorithm called Expectation-maximization (EM) based interacting multiple model (IMM) - EM-IMM algorithm is proposed in [51]. The IMM is capable of identifying states in jumping dynamic models corresponding to various vehicle driving status, while the EM algorithm is employed to give the maximum likelihood estimates of the unknown parameters. Compared to the conventional single model Kalman filter based navigation, the proposed algorithm gives improved estimation performance whereas the land-vehicle drives at changing conditions. Figure 3 presents a sensor integration scheme using a Kalman filter.

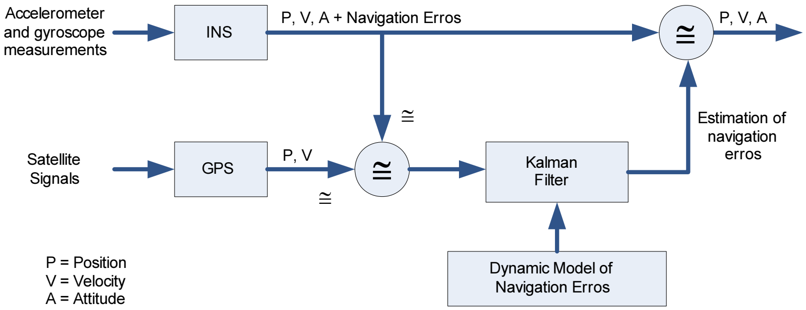

Accelerometer and gyroscope measurements
INS
P, V, A + Navigation Erros
≅
P, V, A
Satellite Signals
GPS
P, V
≅
≅
≅
Kalman Filter
Estimation of navigation erros
Dynamic Model of Navigation Erros
P = Position
V = Velocity
A = Attitude

Figure 3. Kalman-filtering scheme for INS/GPS integration

A different approach regarding the same problem is presented in [53], where the GPS/INS integration method is based on artificial neural networks. The proposed method suggests two different architectures: the position update architecture (PUA) and the position and velocity update architecture (PVUA). Both architectures were developed utilizing multilayer feed-forward neural networks with a conjugate gradient training algorithm. Several practical tests showed the best performance of these methods when they are compared with GPS/INS integration conventional techniques. Figure 4 presents the PUA and PVUA neural network configuration.

An analysis of the integration of an automotive forward-looking radar system (FLRS) sensor with a near object detection system (NODS) and a night vision system with infrared sensors is presented in [60]. The FLRS sensors provide an accurate depiction of the roadway environment within a distance of 100-meter radius. On the other hand, the NODS are short-distance radar sensors that are strategically positioned around the car in order to detect obstacles in general. The integration of both millimeter-wave radars and IR sensors explores the best properties of each for automotive applications.

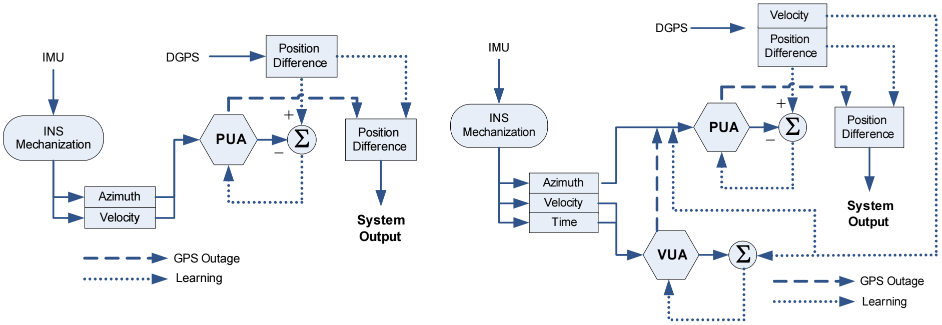


Figure 4. PUA and PVUA system configuration

2.3. Inter-vehicular Communication and ITS

In order to allow several autonomous vehicles to travel simultaneously, there are two ways to assure a certain level of safety. The first one is the use of sensors such as video cameras, radars, etc., aiming at perceiving the other cars. This type of solution simulates the behavior of human beings when they are driving. The second way is the use of aerial communications to attain information regarding the navigation of each car and of the roadway itself and distribute this data among the cars.

Therefore, the main objective when a communication system is used is to transmit information such as position, speed, directions and navigation decisions of the vehicle among the cars that travel around a given covered area. In addition, this communication network enables the remote control and supervision, and also the distribution of useful information to an external host if it is necessary.

This communication system must also meet the following requirements: wireless transmission, real time operation, connecting moving transmitters and receptors, having low errors rates and being able to manage a large number of users.

Several wireless communication technologies are considered for the ITS application. The solutions that allow two-way communication involve Infrared communication, RF, satellite communication service, mobile phone communication (GSM, GPRS and CDMA) and RF data network. Another possibility is the use of wireless broadcast such as FM sideband. In spite of being one of the technologies that present the lowest implementation costs, the communication takes place in just one way and the coverage is limited to the areas covered by FM radio stations. A detailed comparison regarding the several wireless technologies for ITS application is presented in [62-64].

Shladover et al [66], researchers of the PATH project, analyze the possible platoon intra-vehicular communication ways and propose the use of an infrared link to this application. By using an optical transmitter in the rear part and a receptor in the front part of each vehicle, each receptor is able to receive information of the vehicle that is ahead. In [4], the same group of researchers implements a solution based on wireless communication. The used wireless technology is the IEEE 802.11b standard because it is an off-the-shelf commercially ready solution. With this communication system the vehicle communicated periodically, within intervals of 50 ms, at a distance of up to 1 mile traveling at a relative speed of up to 60 mph.

There is still not a defined standard for the architecture of the communication in Intelligent Transport Systems [61]. Although several proposals have been presented, defining this standard is not such an easy task. As it was observed in other applications, in industrial communication networks for example, several proprietary systems are developed before the definition of a communication standard. As a result, different manufacturers develop communication systems that do not have inter-operability.

In 1999, the US Federal Communication Commission (FCC) allocated the 5.850 - 5.925 GHz frequency band to ITS applications. In order to assure its inter-operability, short distance communications will follow the physical and medium access control levels of ANSI/IEEE 802.11a with small modifications [65]. This standard will be used for telecommunication and information exchange between the vehicle and the highway systems.

3. Computer Vision Systems

The major objective of using computer vision in autonomous vehicle is to guarantee efficient autonomous navigation. It means that if the algorithm which extracts information from the environment does not work well, then perception robustness will not work well either. Also the system must be able to recognize any type of weather, daylight or dark, raining or sunny. Speed and consistency are the features which generally are used to evaluate such control system. Complex systems may lead to an error situation once it has to compute heavy processes many times, so it is better running more simple algorithms at higher speeds.

A control computer vision begins with a digitized image which is acquired by a CCD camera (monocular system) or two or more cameras (stereo system) installed on the controlled vehicle.

Once obtained the image from the environment, the main objective of the vision control system is to find out on the image, the path boundary location in the image coordinate system. This can be done by using one of two available methods [24]: edge detection or pixel intensity analysis. In both cases, it is necessary: find and locate a line in the image; eliminate noise or ignore noise and reflections (mainly those caused by the sun light); reduce false detection and reject obstacles that might be confused with the course boundary.

In order to obtain the lane in the image, it is necessary to apply recognition methods, also known as segmentation, which could use the thresholding idea. Thresholding methods can use two ways to run: global and local. In the first case the image is divided in only one value of threshold. In the second case, sub images have its own value of threshold. Global threshold is recommended for images illuminated by uniform distribution of light. So, global thresholding is the easiest way to compute image segmentation when a few regions are interesting on the image. However, multilevel threshold is used when many regions on the image are to be processed and more complicated becomes the process. The thresholding procedure needs a selected value for the threshold. In a controlled environment this value can be obtained statically, by several measurements over that environment. However, outdoors environment which operate in natural light and several changes, like fog or rain, the threshold value is more difficult to be obtained, so, dynamic methods for selecting the threshold intensity must be evaluated.

By using computer vision analysis to environment features recognition, some steps must be done in order to obtain good results in this process: image pre-processing, image segmentation, environment identification.

Most of the process image pre-processing deals with gray level images, which are easy to work, less expensive in time and computer memory and there many methods available to be used. In the case of features environment extraction, applied to control vehicles, sometimes could be interesting in finding out differences between lane marking and road, which generally are colored or signs along the road, which are used to guide the vehicle. In those cases it is recommended to use colored image processing, which is more expensive but keeps more robust detection information.

During this phase, also is necessary to work on the image in order to clean up all its content from noises which come from bad illumination or poor image acquisition. The image noises come from spurious and high intensity pixels that are not part of the image. Depending on the environment conditions, these noises can be quite significant, so, they must receive a special treatment on the image pre processing step. To clean up the image two kinds of filters could be used: one to remove groups of pixels either too small or large to be part of the lines which are part of the image. In both case convolution mask are used in order to perform these tasks.

In the next step, image segmentation, the entire environment is to be reached by the computational method. In this case, the objective is to identify lanes and other things, like cars, signs, road borders, etc.

In general, the success of a lane or obstacle detection process is guaranteed if the marks on the lane are clear among other parts of the image, like noise or other objects. In this case, a simple detector could be used to detect it, but if the marks have low contrast, they are dirty or partially hidden by anything, the recognition method must use more prior knowledge about the road, and then it will take longer to get the results. The more features the system uses to discriminate environment scenes the more parameter it needs to get the real situation. These parameters could be color, texture and shape.

Finally, in the environment identification step, all elements on the image are to be recognized, like the vehicle itself and the obstacles it. This is one of the most difficult steps because there is no such method which solves all kinds of problems and neither structures or models of obstacles. Each case is a particular one. All the environment variables must be considered in order to obtain a good performance from the recognition system. Bayes rules are always considered here in order to decide what family's data some object belongs to.

3.1. Lane Detection

Lane detection is the problem of locating lane boundaries. Many algorithms have been developed at this issue until now. A good list of them is found in [67], [68], [69], [70], [71]. Usually different lane patterns are used; they are solid or dashed, in white or yellow colors, straight or curved lines, two dimensional or three dimensional techniques, etc. An automatic lane detector should be able to handle both straight and curved lane boundaries, the full range of the lane markings, either single or double and solid or broken, in pavement edges under a variety of types, lane structures, weather conditions, shadows, puddle, stain, and highlight.

Tracking of road contours on high speed roads can be automated detected by using Hough Transforms - HT [72]. The road contour of the current lane in the near field of the view are automatically detected and tracked.

In general, lane detection algorithms focus on local image information, like edges, pixel gray levels, and often it fails if the initialization is performed too far away from the expected solution. Some lane detection algorithms require from the operator to provide the initial estimate of the road location, while others require the specific road structure scene, such as straight road or the first road image.

In [20], the Springrobot, is an experimental autonomous vehicle that is equipped with computers, various sensors, communicating device, remote controlling, automatic braking, accelerating, and steering capabilities. Its main target is the development of an active safety system that can also act as an automatic pilot for a land vehicle on structured highways, unstructured road, or unpaved road in rural or city areas. The ability to perceive the environment is essential for the intelligent vehicle. It has been proven that vision sensor is most effective after decades of research, so the perception system of the Springrobot is mainly based on machine vision, combined with others sensors, such as an infrared sensor, radar, laser, differential global position system, accelerometer, gyroscope, etc. To reduce the high computational cost associated with the original input image, the size of the input image is reduced.

Although many features could be used to track the vehicle in a sequence of images, only the 2D bounding box around the vehicle could be used in this case. When analyzing the vehicle image, the system can detect many edges on all sides where the image changes from the vehicle to the background. As it is a linear analysis of searches, Kalman [73] filter can be used in order to assure: extraction of useful information from the tracking process; improving tracking performance by having better expectations of how the tracked vehicle will behave; conveniently integration to the additional measurements provided by the vision system.

The HT is an ordinary method used to extracting global curve segments from an image. This method is able to detect any arbitrary shape undergoing a geometric transformation in an image. However, increasing the number, range, or accuracy of the parameters of the geometric transformations may lead to high computation efforts, which are practically not controllable. Another and more efficient method is presented, the RHT which offers a different approach to achieve increased efficiency in the detection of analytic curves, whereby it has higher parameter resolution, infinite scope of the parameter space, small storage requirements and high speed in relation to the HT method.

In [24] Christine is an autonomous vehicle based vision system which uses simple dynamic thresholding, noise filtering and blob removal in order to accurately identify courses or roadway boundaries. This system is like DARPA, it makes part of a Robotics competition, which holds autonomous vehicles capable of outdoor navigation contained continuous or dashed lines on grass or pavement. An intensity based line extraction procedure has been used in this project in order to effectively to determine when there is no line in the image, for instance when the vehicle is out of the camera's vision field. By counting the number of pixels that occur above the threshold, it is possible to determine if no line is present in the image. The presence of a sufficient number of pixels above the threshold, however, it does not guarantee that there is a usable boundary line in the image. The idea is to count all the pixels and to determine if a line could not be in the image.

Post extraction noise reduction is used once the binary threshold finishes and at this time the image should contain a collection of points that correspond to the image's most intense continuous regions. It cleans up the image by removing spurious pixels that were above the threshold value but do not indicate a line, but should leave a collection of pixels of generally consistent shape and density that compose a line.

In [74] Ralph vision System helps automobile drivers steer, by processing images of the road ahead to determine the road's curvature and the vehicle's position relative to the lane center. The system uses this information to either steer the vehicle or warn the drive if some inappropriate operation is done. In order to locate the road ahead, the Ralph system first re-samples a trapezoid shaped area in the video image to eliminate the effect of perspective and then uses a template based matching technique to find a parallel image featuring in this perspective free image. These features can be as distinct as lane markings, or as subtle as the diffuse oil spots down the center of the lane left by previous vehicles. The information is used by the system to permit it a new adaptation to the environment tracking. A Kalman filter has been used in order to implement the iterative application of Bayes law under Gaussian white noise assumptions for both dynamic and measurement noise. This way, accurate estimates of vehicle motion provide to the system better information for controlling it. So, this system can be able to better recognize discrete events for what they are such as, lane changes or obstacles avoidance maneuvers of the car ahead. Using the accurate measurements of relative position and speed of other vehicle given by the system, the system can predict which is ahead and construct a tactical plan to maneuver through traffic.

In [22] to precisely track the transition of target points, the points must have some characteristics in the image, which can be used to detect their positions both in lateral and longitudinal directions. In order to simplify the image processing, only the lateral features are considered in this process and so the image is scanned from right to left on the image plane. White continuous and broken lines are considered as the candidates of the target points; only the brightness level of the image is used to detect target points. To detect those points, a window with a lateral width is set at determined point on the image, which is the predicted position of the target point calculated from the vehicle movement. When the system detects a brightness change in the window position, this position is determined as the target. The calculation of the transition of the target points are done by the perception subsystem based on the information from the vehicle control system, which is its travel distance and steering angle.

In [75], BART is vision based autonomous vehicle consisting of an image processing system, a recursive filtering and a driving command generator which includes an automatic steering and speed control in order to provide safety hide between the lead vehicle and its follower. The image processing module obtains the position of the lead vehicle by using the relationship between the tree dimensional coordinates of a distinguishable tracking feature on the back of the lead vehicle and the corresponding coordinates of the left and right images representing the projection of that feature through a stereo camera model. Contours, shapes, or color patterns can be utilized as tracking features. However, since the vehicle is required to run in real time, time consuming image processing operations must be avoided. One way to provide this economy of time is using a spot light in order to simulate the coordinates of the leader vehicle. These experiments have been conducted considering straight line roads, lane changing treatment and curved line roads.

In [76] a control method of autonomous vehicle using visual servoing is used to guarantee its performance. Visual servoing is an approach to the control of robots based on visual perception, involving the use of cameras to control the position of the robot relative to the environment as required by the task. This idea can be used in issues related to computer vision, robotics, control, and real time systems. To control an autonomous vehicle by visual servoing, marks for autonomous running, namely landmarks are necessary. In general, the landmarks are specified by an operator at the teaching step, which is carried out before autonomous running was effectuated. However, this task can be automated by performing the teaching operations and improving the robustness of autonomous running. By its production system, the operator can perform the teaching operations without an expert on visual servoing and the vehicle can adapt itself to disturbances while autonomous running.

The advantages of this method is that, it is free from the self position measuring error as when using internal sensors, being free from the need to prepare data on the surroundings, and being free from the effects of calibration error of the camera on running control compared with other methods using visual sensors.

In [77] other technologies for this subject are described.

4. High Level Control Case-Selection

Despite the application, the main purpose of an autonomous vehicle is to safely navigate from its current position to a given target. It would not be an easy goal even if the environment was completely static and well known. Therefore, in real work applications, a series of situations associated with different types of solutions could happen, demanding the capability of finding and switching the current control strategy. For instance, if an autonomous car is navigating inside lane marks and an obstacle announcing a possible collision is detected, the current control (lane keeping) must be changed, in order to avoid impact or, at least, mitigate it. Hence, for a fully autonomous vehicle, the decision-making system is a main issue. For each detected environment situation, one or more control strategies must be executed. To determine which one is more suitable for a specific moment, researchers tend to use a finite-state machine. Reference [2] presents a driving assistance system intended to be used under controlled conditions. When more dangerous scenarios are detected, an alarm requests the driver to take over the vehicle control. Reference [7], however, present a fully autonomous car in the Darpa Challenge, where all decisions are made and executed by the vehicle.

The most common situations in an autonomous driving are: local and global navigation, road following, car tracking, vehicle passing, lane changing, stop and go traffic, collision avoidance and mitigation, maneuvering, and parallel parking. These topics, along with self localization and the high level control execution framework, are presented next.

4.1. Autonomous Localization

An important issue involving all methodologies of intelligent navigation is localization. Since autonomous vehicles are migrating from well behaved tests in laboratories to real-world scenarios, the techniques involved in such applications had to be improved. Although landmarks, beacons or certain pre-defined objects can still be used in non-controlled outdoor situations, they are definitely insufficient. Therefore, a series of information provided by a large range of sensors must be fused into reliable localization. However, these sensors accumulate errors providing imprecise and even ambiguous measurements. In this case, the Bayes filter [8] is used to estimate the current position. It assures that it is possible to identify a proper way to determine the belief in a given hypothesis with imprecise evidences. There are two ways of representing a vehicle's belief depending if the goal is to determinate its local or global positioning. Local recognition demands not only the determination of its relative position in relation to a reference point, but also the indication of the localization of objects, obstacles, traffic lights, etc. This detection can be based on the sensorial identification of certain elements in the environment, and may process this information in two different ways: a) identifying the precise position, if the localization of the identified elements are well know, or b) use multivariate Gaussian densities that uses Kalman filter methods to clean up the initial data and solve the problem under an uncertain approach. However, if the pursue is global localization, Markov discretization is the right technique. It breaks the state space by mapping the topological structure of the environment [9]. Although Markov discretization is a powerful methodology, it is also time-consuming. To avoid the extra computational effort, reference [10] asserts that the vehicle's belief can be represented by of weighted random samples of robot positions and constrained based on observed variables. Moreover, by using Monte Carlo and condensation methods, it is possible to indicate very efficiently the global localization.

4.2. Global and Local Navigation

Global navigation is an old and relatively easy issue. It involves the knowledge of geospatial data of the desired areas and an algorithm based on graph search. The most common, and also very reliable, algorithm is the A*, which searches the solution tree for an optimized path based on a given heuristic function such as distance traveled, security, time spent, etc. Examples of using A* can be found in detail in reference [7]. Another methodology, Case-Based Plan Reuse, solves new problems by retrieving and adapting previously formed plans, as shown in [11].

While global navigation can be very easy, local motion planning is not. In such situation, the vehicle does not know the environment in advance. Hence, it must be able to simultaneously execute, in real time, learning, reasoning and problem solving techniques.

The artificial potential field approach and its variations are among the well-known techniques which have been developed for local navigation [12-17]. The literature also shows techniques based on Bayesian Networks [7], Fuzzy Logics [18], Direct Formulation Extraction [19], and several others based on video estimation [20 - 26]. Figures 5a and 5b [20] show a local path detection using different video techniques.

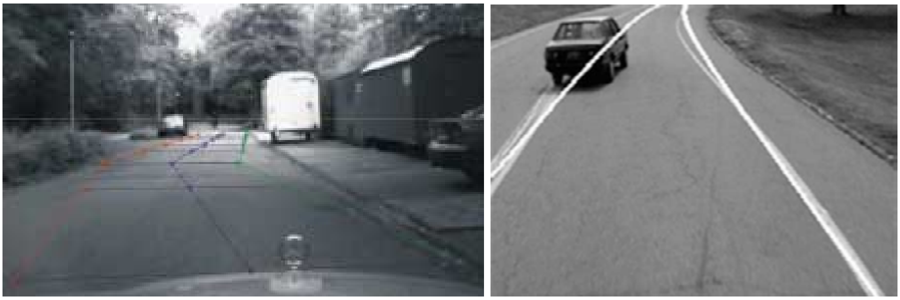


Figure 5. (a) local path planning using Bayesian networks (source [25]), (b) a detected lane used to identify the path (source [26]).

4.3. Vehicle Following and Passing, Stop and Go, Collision Avoidance and Mitigation

In real environments, it is common for an autonomous car to find itself in a situation where the local navigation must be constrained by external influence, i.e., obstacles, lights signals, etc. Hence, once these influences have been correctly identified, a specific type of algorithm must be executed in order to safely control the vehicle. The literature reports several researches focusing on these applications. Reference [27] shows a neural-network approach to vehicle following. References [15-16] use the already cited potential field methods, which are based on creating a surface with a sink representing the goal and sources representing the obstacles. While [15] uses the original methods, [16] changed the formulation, in order to avoid trapping situations.

Reference [28] was developed based on elastic bands, yielding good results but poor computational response.

Figure 6 shows three different situations present in a real environment: (a) obstacles identification – in this case trucks and pedestrians, (b) a car following system, and (c) a stop and go situation with collision avoidance, obstacles and traffic lights identification.

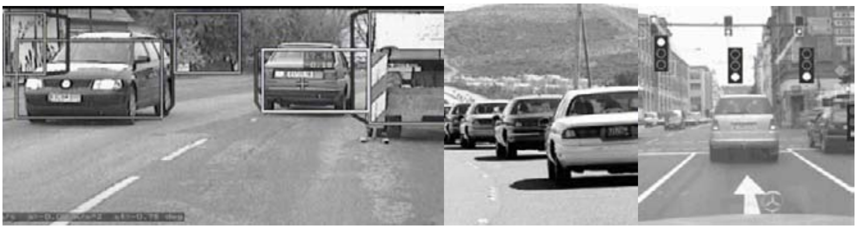


Figure 6. (a) obstacle detection and avoidance, (b) car following in a platoon formation, and (c) traffic light and obstacle detection in a vehicle following state, involving situations of stop and go and collision avoidance. {a from source [38] ; b and c from source [6]}

4.4. Maneuvering and Parallel Parking

One of the most appealing, and even commercial, autonomous driving situation is parallel parking. It has been attracting a lot of attention from researchers and from the industry, leading to a series of methodologies. Basically, there are two main approaches to defining a parallel parking motion control: a mathematical model representing the physical equations of the vehicle [29,30] or fuzzy rules given by a driver expertise [31-33]. Figure 7 shows the BMW parallel parking assistance.

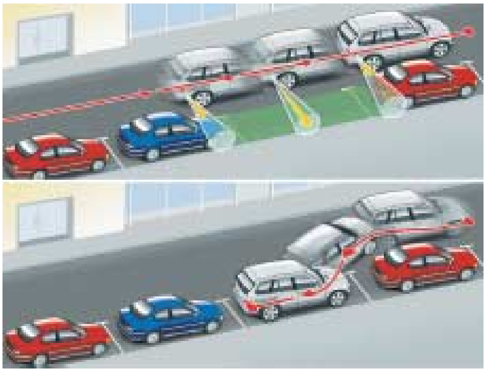


Figure 7. BMW Connected Drive Project (source: www.bmw.com)

4.5. Control Execution

The execution of this type of decision demands a highly precise response and integration between this layer and the low level control. It is necessary to correctly identify a given situation, provide the control strategies, and know when the situation is over. The situation is more challenging when it is taken in account that different computers may be responsible for each desirable control. This leads into a situation where several systems must provide real-time data and information, strategies, and controls, performing actions in the vehicle. In that case, distributed real-time software is required. To deal with this challenge, Reference [7] used the Real-Time Control System Library [34] and Reference [35] used value-based scheduling system. However, when commercial cars are the focus, several studies about dealing with real-time information through the Controller Area Network (CAN) are present in the literature [36-37].

5. Proposal of a High-level Control based on Learning and Optimization

5.1. Background

As mentioned in section 1 and stated in section 4, the architecture of intelligent vehicles typically presents high-level controllers, which means powerful algorithms based on Fuzzy Logic, Genetic Algorithms, Neural Networks, Machine Learning, Planning and other Artificial Intelligence Techniques.

The design of fuzzy controllers depends on the expert's personal experience. In this case, one must determine the association rules between the inputs and outputs of the system to be controlled. However, in some applications the expert knowledge can be imprecise or may not be enough to build such controllers. In scenarios like these, concepts about Machine Learning and Data Mining can be used to analyze an initial data set (IDS) and generate both fuzzy rules and membership design.

The literature presents several methodologies to extract fuzzy rules from numerical data [78], [79]. A commonly used approach is the construction of fuzzy rules using a decision tree built from data samples. Probably, the most popular algorithms are ID3 [78] and C4.5 [79]. In [80] a classification-rule discovering algorithm integrating Artificial Immune Systems (AIS) and fuzzy system is proposed. The algorithm consists of two parts: a sequential covering procedure and a rule evolution procedure. Each antibody (candidate solution) corresponds to a classification rule. The classification of new examples (antigens) considers not only the fitness of a fuzzy rule based on the entire training set, but also the affinity between the rule and the new example. Although all these methods presented very good results, they are based on the principle that each rule of the inference process must be built by individually learning each part of the antecedent. This strategy generates a huge solution space, and the computational time necessary to swap this domain is very high. One approach used to narrow the solution space is presented in [81] where, from standard memberships, the algorithm uses a simple logic to assemble rules capable of representing a given data set. This logic is based on associating the input and the output over the standard memberships where each entry of the original data set may generate a rule. The output of this step is a repertoire of rules, which will be treated to eliminate inconsistency. This method also presents good results and is able to deal with large data sets. However, this approach assumes standard and fixed memberships and the final rules repertoire may be quite inexpressive, because it uses several considerations to avoid inconsistency and redundancy.

To improve the approach presented in [81], this work developed a method using a Co-Evolutionary Gradient-Based Artificial Immune System with two different types of populations: one is responsible for optimizing the memberships and the other for learning the rules. The optimization process begins by generating a population of antibodies where each one is responsible for holding information about the positions and shapes of all memberships associated with the system's variables. The learning process starts by analyzing the initial data set and building an available rules table (ART). The ART is assembled using the main idea provided by [81] but without the approach of avoiding inconsistency. Therefore, it contains all possible rules necessary to represent the data set. Continuing the process, the second population of antibodies must decide what rules in the ART should be used for a better representation of the IDS. These two populations co-evolve together until the end of the process. However, even though the populations have a strong relationship and the same goal, to accurately represent the data set, they may present conflict of interest, which means that improvements on one may produce negative impacts on the other. To avoid conflicts like this, a full Pareto optimization [82] is used and one population will only evolve if it does not jeopardize the other.

In order to evaluate this methodology, a scenario of automatic parallel parking will be used, where a fuzzy system for autonomous maneuvering will be learnt by a data set of actions taken by a human driver. This article is organized as follows: Section 2 shows the main methodology, Section 3 the co-evolutionary gradient based artificial immune system, Section 4 shows the results and, finally, the conclusions are discussed in Section 5.

5.2. Generating a Fuzzy System from a Numerical Data Set

Suppose a given numerical input and output data set acquired from a given environment. The goal is to automatically generate a fuzzy system that represents the relation between the inputs and outputs of this application. Such system is composed of two types of elements: membership functions and rules. Membership functions transform real variables from/to membership degrees and rules express the knowledge domain based on these indexes. Membership functions may be present in different sets of numbers, shapes and positions, and finding the best configuration depends on the systems data and rules. The literature presents several works dedicated to generating fuzzy systems from data [83-85] using three different approaches. One approach gets the rules from an expert and uses population-based algorithms to optimize the memberships [86], the second approach provides the memberships and learns the rules [81], and, finally, some algorithms uses two different populations to learn both rules and memberships [83]. As the optimization of membership functions and the rule inference process are dependent of each other, they should both be modeled together to reach the best representation. Adopting this approach, we present an algorithm, the CAISFLO, based on two different sets of populations co-evolving together to find the best representation of a real application where one set is responsible for optimizing the memberships while the other for learning the rules. However, even though they have the same goal, i.e. to generate an accurate representation of a data set, changes that improve one population may destroy the other. For instance, if a new membership function is added to a variable, the current rules may not represent the system anymore and the rules population should be restarted. Thus, taking this action could reduce the fitness, and losing track of the evolutionary process. To avoid this situation, a full Pareto optimization [82] is adopted, meaning that improvements in one population will be only allowed if it does not jeopardize the other.

As already stated, fuzzy rules are strongly connected and depended on membership functions (MF). It is impossible to find any rule in a data set if these functions have not been defined. Thus, the first step of assembling a fuzzy system is to generate a population of membership functions *fmPop*={*fAb*$_1$,...,*fAb*$_n$}. For each individual *fAb*$_i$, a new population *rPop*$_i$ = {*r*$_i$*Ab*$_1$,...,*r*$_i$*Ab*$_m$} responsible for learning the inference rules is created and evolved. After the rules are learnt, each individual *fAb*$_i$ of the first population will have its memberships optimized to enhance the accuracy. This process is shown in Figure 8 and continues until the end of the simulation.

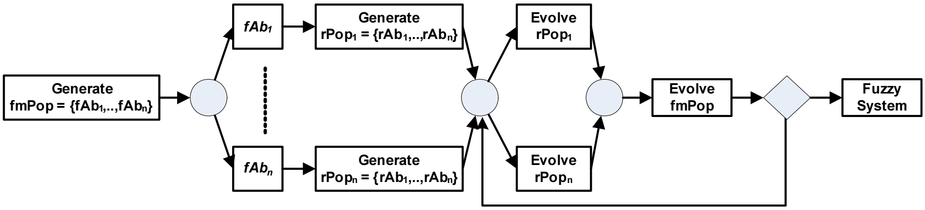

Generate fmPop = {fAb₁,..,fAbₙ}
fAb₁
fAbₙ
Generate rPop₁ = {rAb₁,..,rAbₙ}
Generate rPopₙ = {rAb₁,..,rAbₙ}
Evolve rPop₁
Evolve rPopₙ
Evolve fmPop
Fuzzy System

Figure 8. Co-Evolutionary process.

As the main purpose of this approach is to correctly represent a real system given a data set, the fitness of antibody *fAb*$_i$ is given by

$$fitness(fAb_i) = \sum_{a=1}^{Ne} (fAb_i(ipDS_a) - opDs_a)^2 + Penalty(ipDS_a) \tag{1}$$

where *Ne* is the number of entries in the data set (DS), ipDS$_a$ represents the input variables vector in entry *a* of DS, opDs$_a$ represents the output value of entry *a,* and *Penalty(.)* is a function that returns a very large value (i.e. 10^{99}) if *fAb*$_i$ does not have any rule to deal with *ipDs*$_a$.

The main idea of how these populations individually work are shown next, whereas Section 5.4.1 presents the methodology for learning fuzzy rules and section 5.4.2 shows the membership optimization.

5.3. Learning Fuzzy Rules

In order to facilitate and emphasize the explanation of the proposed method, suppose a numerical data set with two input variables (x_1 and x_2) and an output variable (y). The given members of the data set are represented as:

$$(x_1^1,\ x_2^1;\ y^1),\ (x_1^2,\ x_2^2;\ y^2),\ldots,\ (x_1^n,\ x_2^n;\ y^n) \tag{2}$$

The first step of the proposed methodology consists of generating a table of available rules (ART). This table contains all possible rules from a given data input taking into consideration a set of membership function configuration. The ART is generated from the division of the input and output range into fuzzy regions, similarly to step 1 of the proposal made by [81]. Each domain interval, i.e., the range comprehended between the minimum and the maximum values of a data set variable must be divided into 2N+1 regions where N can be different for each variable.

Figure 9 shows the division of the interval domains into fuzzy regions. In this case, the domain interval of the input variable x_1 is divided into three regions (N=1), the input variable x_2 into five regions (N=2), and the output variable y also into five regions (N=2). After that, according to [81], each data set entry may generate a single fuzzy rule based on the highest membership degree of every variable. For example, Figure 9 shows that x_1^1 has a membership degree of 0.6 at S1, 0.4 at M and zero at the other regions. Likewise, x_2^1 has a degree of 1 at M and zero at the other regions, whereas y^1 has a degree of 0.8 at L1 and 0.2 at M. Thus, this entry generates the following rule:

$$(x_1^1,\ x_2^1;\ y^1) \Rightarrow \text{IF } x_1 \text{ is S1 AND } x_2 \text{ is M, THEN } y \text{ is B1} \Rightarrow \text{Rule 1} \qquad (3)$$

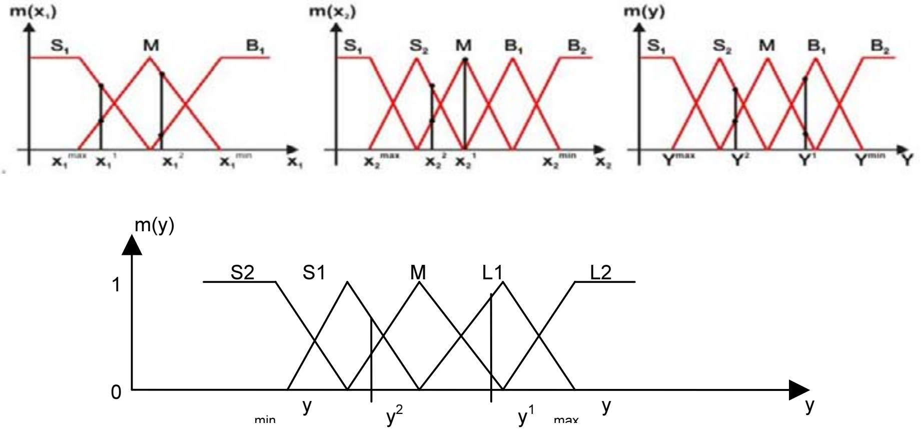


Figure 9. Division of the input and output space into fuzzy regions.

It is well-known that the numerical output value of a fuzzy system depends on the activated membership functions with its respective degree. It is important to highlight that all of the activated membership functions with a higher or a lower degree contribute to the output value calculation. Therefore, generating fuzzy rules based on the highest membership degree, as previously shown, means a simplification of the problem. Instead of generating rules based on membership degrees, our proposal is to generate a set of rules combining all fuzzy regions that have activated membership functions, whatever the membership degree. Thus, instead of generating a single fuzzy rule according to what was shown in Eq. (**3**), the given member of the data set (x_1^1, x_2^1, y^1) will generate:

$$\begin{array}{l} (x_1^1,x_2^1;y^1) \Rightarrow \text{IF } x_1 \text{ is S1 AND } x_2 \text{ is M, THEN } y \text{ is } B_1 \Rightarrow \text{Rule 1} \\ (x_1^1,x_2^1;y^1) \Rightarrow \text{IF } x_1 \text{ is S1 AND } x_2 \text{ is M, THEN } y \text{ is M} \Rightarrow \text{Rule 2} \\ (x_1^1,x_2^1;y^1) \Rightarrow \text{IF } x_1 \text{ is M AND } x_2 \text{ is M, THEN } y \text{ is } B_1 \Rightarrow \text{Rule 3} \\ (x_1^1,x_2^1;y^1) \Rightarrow \text{IF } x_1 \text{ is M AND } x_2 \text{ is M, THEN } y \text{ is M} \Rightarrow \text{Rule 4} \end{array} \qquad (4)$$

At first, several conflicting rules, i.e., rules that have the same antecedent (IF part) but a different consequent (THEN part), are generated when this procedure is adopted.

In order to solve this problem, the ART is built in a way that each line represents a unique antecedent and each column of the consequent part represents the output membership functions. Each antecedent will be associated with a certain number of activated output memberships, signed with a digit "1" in the ART.

As an example, Table 2 shows the generated ART from the given members (x_1^1, x_2^1; y^1) and (x_1^2, x_2^2; y^2) taken from Figure 9. In this case, the output variable y is represented by five columns, S2, S1, M, B1 and B2, that correspond to their membership functions. Here, just two given members of the data set have generated eleven potential fuzzy rules, whereas, in the method based on membership degree, only two potential fuzzy rules are generated. It is important to note that not all potential fuzzy rules will be used in the final solution. Instead, a combinatorial optimization using the GbCLONALG algorithm will determine the best repertoire to be used. This process will be explained in Section 5.4.

Table 2. Available Rules Table – ART

Antecedent	Consequent - Output y				
	S2	S1	M	B1	B2
IF x_1 is S1 **AND** x_2 is M **THEN**	0	0	1	1	0
IF x_1 is M **AND** x_2 is M **THEN**	0	1	1	1	0
IF x_1 is M **AND** x_2 is S1 **THEN**	0	1	1	0	0
IF x_1 is B1 **AND** x_2 is S1 **THEN**	0	1	1	0	0
IF x_1 is B1 **AND** x_2 is M **THEN**	0	1	1	0	0

5.4. The Gradient-Based Artificial Immune System

The Artificial Immune System (AIS) intends to capture some of the Nature Immune System (NIS) within a computational framework. The main purpose is to use the successful NIS process in optimization and learning [87]. As every intelligent-based method, the AIS is a search methodology that uses heuristics to explore only interesting areas in the solution space. However, unlike other intelligent-based methods, it provides tools to simultaneously perform local and global searches. These tools are based on two concepts: hypermutation and receptor edition. While hypermutation is the ability to execute small steps toward a higher affinity Ab leading to local optima, receptor edition provides large steps through the solution space, which may lead into a region where the search for a hAb is more promising.

The technical literature shows several AIS algorithms with some variants. One of the most interesting ones is the GbCLONALG algorithm presented in [88]. The main statement of GbCLONALG is that progressive adaptive changes can be achieved by using numerical information (NI) captured during the hypermutation process. There are several possible ways to capture this information. The one used in this article is the tangent vector technique, because it treats the objective function as a "black box", where small disturbances are individually applied over each dimension of the input vector yielding the vector

$$\mathrm{TV}\left(\mathrm{f}\left(\mathrm{x}_1,\ldots,\mathrm{x}_n\right)\right)=\begin{bmatrix}\dfrac{\mathrm{f}(\mathrm{x}_1+\Delta\mathrm{x}_1,\ldots,\mathrm{x}_n)-\mathrm{f}(\mathrm{x}_1,\ldots,\mathrm{x}_n)}{|\Delta\mathrm{x}_1|}\\ \vdots \\ \dfrac{\mathrm{f}(\mathrm{x}_1,\ldots,\mathrm{x}_n+\Delta\mathrm{x}_n)-\mathrm{f}(\mathrm{x}_1,\ldots,\mathrm{x}_n)}{|\Delta\mathrm{x}_n|}\end{bmatrix} \tag{5}$$

where: *n* is the number of control or input variables, *f(.)* is the objective function to be optimized; $x_1,\ldots,x_n$ are input variables; and Δx_k is a random increment applied to x_k.
This approach enhances the algorithm efficiency and enables a well defined stop criterion. The complete algorithm is shown in Figure 10 and a pseudo code of the new hypermutation process is shown in List 1 where:
Ab{nps}{n}-structure representing a population of *nps* antibodies and *n* input variables; **Ab{i}{j}**-field representing the input variable *j* of antibody *i*.; **Ab{i}.TV -** field of size *n* representing the tanent vector of Ab{i}.; **Ab{i}.open -** Boolean field representing if antibody *i* is still open for further mutations or not. When a new antibody is generated this field is TRUE. **Ab{i}.α** -field representing a mutation factor.; **Ab{i}.fit** -field representing the fitness.; **F(Ab{i}) -** fitness function.; **cAb{i} -** clone of Ab{i}, inherits all fields of Ab{i}.; **β -** parameter to control the mutation decay in case of improvement failure. ; $\boldsymbol{\alpha}_{\mathbf{min}}$ -minimum allowed step size.

List 1. Pseudo code of the hypermutation process

```
FOR EACH Ab{i} IN Ab{nps}
  IF$_1$ Ab{i}.open == TRUE
      Ab{i}.gr = TV(Ab{i});
      cAb{i}=Ab{i}+Ab{i}.α×rand×Ab{i}.gr;
      cAb{i}.fit = F(cAb{i});
      IF$_2$ (cAb{i}.fit > Ab{i}.fit); Ab{i}.fit = cAb{i}.fit;
      ELSE$_2$
        Ab{i}.α = β × Ab{i}.α;
        IF$_3$ (Ab{i}.α < $α_{min}$ )Ab{i}.open = FALSE; END IF$_3$
      END IF$_2$
  END IF$_1$
END FOR EACH
```

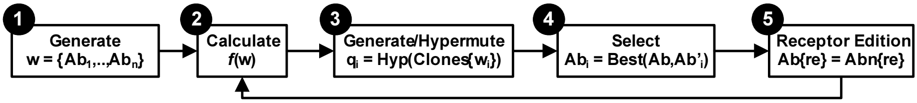


Figure 10. GbCLONALG flowchart

Each step or block of the previous diagram is detailed as follows:

1. Randomly choose a population w = {$Ab_1,\ldots,Ab_n$}, with each individual defined as Ab_i = {$x_1,\ldots,x_j\ldots,x_{nc}$}, where *nc* represents the number of control variables or actions;
2. Calculate the value of the objective function for each individual; this result provides the population affinity for the optimization process;
3. Proceed with the Hypermutation process accordingly to List 1.
4. The bests *nb* individuals among the original *w* population are selected to stay for the next generation. The remaining individuals are replaced by randomly generated new *Ab's*. This process simulates the receptor edition and helps in searching for better solutions in different areas.

Although this algorithm has presented very good results in continuum optimization scenarios [88] it is necessary to adapt its principles to carry out combinatorial search problems. To accomplish that goal, some considerations about the antibodies, the tangent vector calculation and the receptor edition must be taken in account:

- *Antibodies Definition* - An antibody represents a path over a tree search. It starts at the root and, as its evolution occurs new nodes are added to the path.
- *Discrete Tangent Vector Definition* - The clones are responsible for analyzing a given node. Their population expands branches from their parents and the numerical information obtained from this process should indicate the likelihood of finding the best solution following a given path. The index adopted in this article to evaluate this likelihood is

  $$TV = v_i \times \Phi\{\mathrm{v}_1,\ldots,\mathrm{v}_{\mathrm{nLocalbest}}\} \qquad (6)$$

  Where the mean value Φ provided by the *nLocalbest* individual of each branch, and multiplies it by each individual clone value v_i.

- *Receptor Edition* - Changes in the input variables can lead the system into a complete different solution. Thus, during the hypermutation the adopted strategy is to keep the *nLocalBest* clones instead of just one. By using this concept the algorithm allows that a given antibody, which is not well classified at first, may evolve and became the best. To avoid a combinatorial explosion, at each interaction, all the clones are compared and only the *nGlobalBest* are kept alive by the receptor edition.

Applying these concepts to generate a fuzzy system demands to divide the problem in two phases. On the first one, rules generation, the GbCLONALG must select the best sets of fuzzy rules. For this purpose, an initial set of fuzzy memberships (ISFM) is used to analyze the initial data and to generate the ART containing all possible rules. A population of antibodies is created, where each one is a repertoire of possible rules picked from the ART. The population evolves using the tangent-based hypermutation process, selection and receptor edition. The output of this phase is a set of antibodies containing a repertoire of if-then-based rules, representing the initial data according to the initial set of memberships. The second step can be considered as a fine adjustment, where the ISFM is optimized to enhance the accuracy of the fuzzy system. Both phases are better explained next.

5.4.1. Fuzzy Rules Generation

As explained in last section, the methodology used in this work is an improvement of the one first proposed in [81]. It creates a table where each column represents an output and each line represents an antecedent rule. This approach improves the original algorithm in two ways: it works with a number of rules more likely to represent the system, and by implementing the output in columns, it ensures that all the inconsistencies will be avoided. This last statement is due to the fact that only antecedent rules that present different outputs are inconsistence and, and once the algorithm is responsible to chose among the available rules in the table, it is easy to prevent against this situation. The utilization of the GbCLONALG in the present domain problem is shown in Figure 11 and explained next.

Keeping in mind that an antibody is a repertoire of rules, the first step is, given the ART with *nr* entries, to select every available rule, generating *nr* antibodies. The second step is to carry out the hypermutation process and, for that purpose, it is necessary to calculate the Tangent Vector. To do that, a random number of rules will also be randomly picked from the ART, added to a hypermutated clone (i.e. hCab1$_1$) and evaluated. The selection process begins and, using the values of the *nLocalbest* clones of a given antibody, the TV is evaluated according Eq. 6. Then, the *mGlobalbest* clones will generate the new population. The process carries on according to Figure 11.

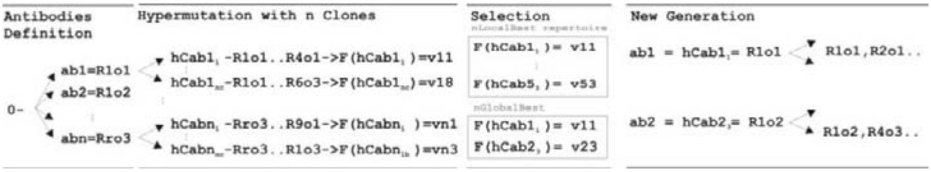
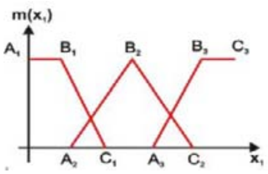


Figure 11. GbCLONALG applied in discrete optimization

5.4.2. Membership Function Optimization

In the membership optimization, each antibody represents a set of vectors containing the coordinates of the membership functions. An example is shown in Figure 12 where Ab = {A_1,B_1,C_1,A_2,B_2,C_2,A_3,B_3,C_3}. The optimization process occurs according the GbCLONALG and the result is the best adjust of each position of each vector. Considerations about limits over the variables are also shown in Figure 12.



where

$A_1 = Min$

$Cn = Max$

$A_1 < B_1 < B_2 < \ldots < B_n < C_n$

$B_K < C_K < B_{k+1}$

$B_K < A_{K+1} < C_k$

Figure 12. Memberships Coordinate

5.5. Application and Results

In order to validate the proposed method, this paper presents an application of automatic parallel parking using a 3D software [89] that allows the reproduction of vehicle dynamics. Some sensors, such as GPS, inertial, ultrasound, infrared and laser, were modeled to perceive the vehicle surroundings. Figure 13 illustrates the used environment, as well as the considered input variables. To build the data set, a parallel parking maneuver was manually carried out by using a joystick. During this process, the input variables X_A, Y_D and θ as well as the OUTPUT value obtained by reading a virtual encoder connected to the steering wheel, are stored in a data set, yielding 256 entries.

To extract a fuzzy system from this data set, capable of reproducing the human control over the vehicle the CAISFLO algorithm was used. The first step is to generate an initial population of memberships and, for each one, build the ART.

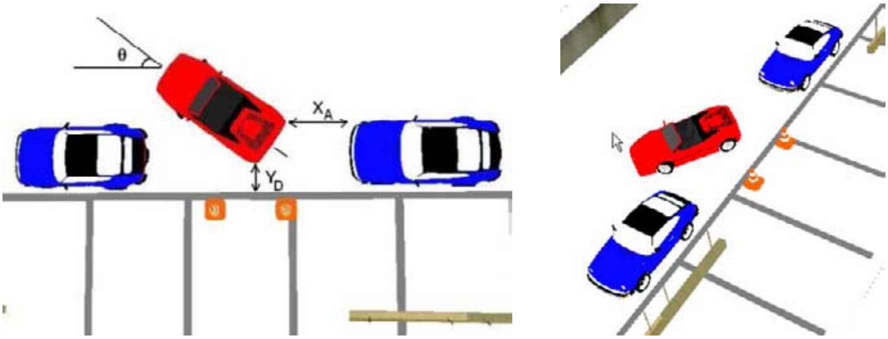


Figure 13. Simulation software and the variables considered in the fuzzy control

To illustrate this process, Figure 14(a) shows the initial membership configuration of an antibody. However, in order to compare the CAISFLO, the configuration was generated according to [81]. For this antibody, the ART is shown in Table 3, where 22 possible rules were considered. If the methodology proposed in Wang&Mendel [81] had been used, just 6 rules would have been generated, as shown in Table 4. As the populations evolve, the memberships and rules start to assume new shapes with better results. Figure 14(b) shows the result of CAISFLO's first generation where it is possible to see that the membership function M_{XA} assumed a trapezoidal shape. Figure 14(c) and Table 5 showed the final result of memberships and the rules, respectively, obtained after 3 generations. It is important to note that the final number of rules obtained from the present proposal have found only 5 rules. Although it was just one rule lower than the method of W&M, the rules are different and have more accuracy, as can be seen in Figure 15 where the y-axis represents the virtual encoder value and the x-axis the entries in the data set. One of the many reasons for that can be explained by comparing the consequent values of Tables 4 and 5. Note that the W&M method used just two consequent memberships (S and B) while CAISFLO used three (S, M and B) providing a smoother transition among the operational states of the system. Also note that, by analyzing Figure 15, the W&M was not able to find any rule to deal with the entries inside the interval (1, 36).

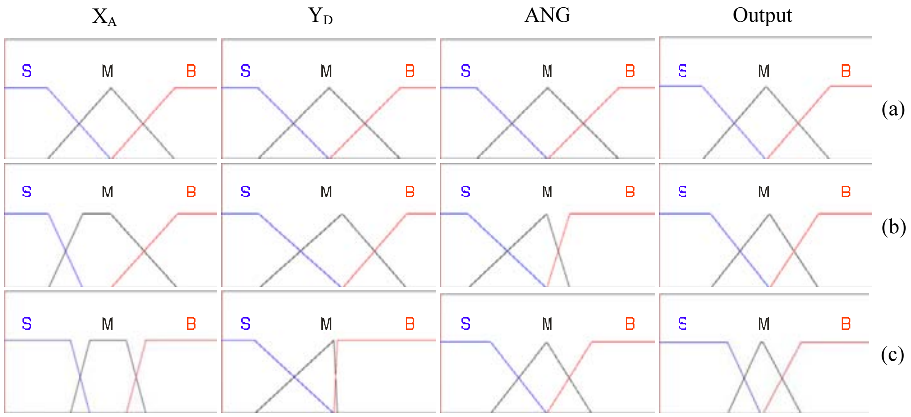


Figure 14. (a) Membership Functions without optimization (b) Optimized Membership Functions, and (c) Optimized Membership Functions after co-evolution.

Table 3. Available Rules Table.

Antecedent	Consequent		
	S	M	B
IF X_A = B **AND** Y_D = B **AND** ANG = S **THEN**	0	1	1
IF X_A = B **AND** Y_D = B **AND** ANG = M **THEN**	0	1	1
IF X_A = B **AND** Y_D = B **AND** ANG = B **THEN**	0	0	1
IF X_A = M **AND** Y_D = B **AND** ANG = B **THEN**	1	1	1
IF X_A = M **AND** Y_D = M **AND** ANG = B **THEN**	1	1	1
IF X_A = B **AND** Y_D = M **AND** ANG = B **THEN**	0	0	1
IF X_A = S **AND** Y_D = M **AND** ANG = B **THEN**	1	1	1
IF X_A = S **AND** Y_D = B **AND** ANG = B **THEN**	1	1	1
IF X_A = S **AND** Y_D = S **AND** ANG = B **THEN**	1	1	0
IF X_A = B **AND** Y_D = S **AND** ANG = B **THEN**	1	1	0

Table 4. Rules obtained from Wang&Mandel

Antecedent	Consequent
IF X_A = B **AND** Y_D = B **AND** ANG = M **THEN**	OUTPUT = B
IF X_A = B **AND** Y_D = B **AND** ANG = B **THEN**	OUTPUT = B
IF X_A = M **AND** Y_D = B **AND** ANG = B **THEN**	OUTPUT = B
IF X_A = M **AND** Y_D = M **AND** ANG = B **THEN**	OUTPUT = B
IF X_A = S **AND** Y_D = M **AND** ANG = B **THEN**	OUTPUT = S
IF X_A = S **AND** Y_D = S **AND** ANG = B **THEN**	OUTPUT = S

Table 5. Rules obtained from CAISFLO

Antecedent	Consequent
IF X_A = B **AND** Y_D = B **AND** ANG = S **THEN**	OUTPUT = M
IF X_A = B **AND** Y_D = B **AND** ANG = B **THEN**	OUTPUT = B
IF X_A = M **AND** Y_D = B **AND** ANG = B **THEN**	OUTPUT = B
IF X_A = S **AND** Y_D = B **AND** ANG = B **THEN**	OUTPUT = M
IF X_A = S **AND** Y_D = S **AND** ANG = B **THEN**	OUTPUT = S

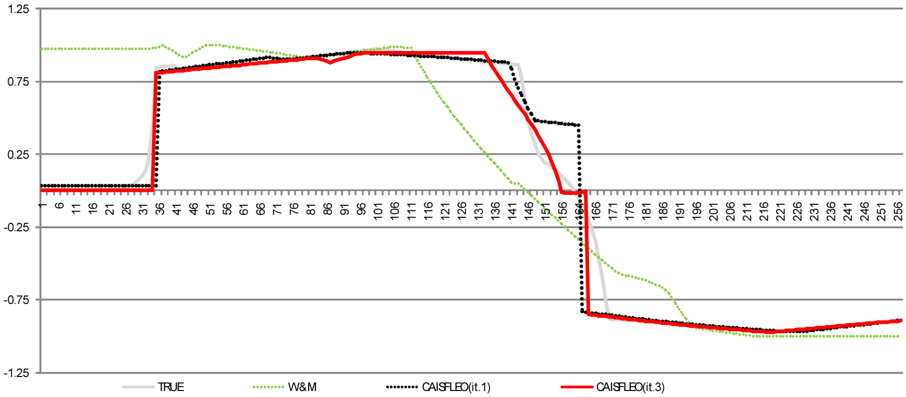


Figure 15. The original (TRUE) output from the data set, the results of the W&M method, and the 1st and final generations of CAISFLO.

6. Conclusion

In this work, we have presented a general overview of autonomous vehicles, main topology and navigation systems. Several articles, some involving competitors of the DARPA Desert Challenge, have been present. It was also demonstrated the growth of this field in the last decade.

The perceptions and acquisition of knowledge on the environment where the intelligent vehicle will navigate can be carried out through different types of sensors. The solutions regarding the instrumentation in intelligent vehicles use mainly DGPS, INS, cameras, radars, laser sensors and IR. Functionally, these sensors can be organized as: route recognition, obstacle recognition or navigation sensors.

As each type of sensor has its own characteristics, advantages and disadvantages, a widely used solution is the use of several sensors and the fusion of the information they provide. Some fusion techniques were presented in this paper. Most of the solutions are based on the use of the Kalman filter. In addition, a solution based on artificial neural networks was also presented.

When the case concerns more than one intelligent vehicle traveling on a highway, besides the on-board sensors, the inter-vehicular communication and the communication with the highway itself are also important. Although there is still not a communication architecture that is fully standardized for this type of application, there is a strong tendency towards the use of wireless communication using the IEEE 802 standard. However, the other communication protocols are yet to be defined.

This work also presented a co-evolutionary artificial immune based algorithm applied to generate fuzzy systems from numerical data. For that purpose, two sets of populations were used: one designed to learn rules and another to optimize membership functions. To avoid co-evolutionary problems, where one population disturbs the evolution of others, a full Pareto optimization paradigm was employed.

As any probabilistic optimization method, the AIS algorithm searches the solution space for good results, implying that the greater the space, the higher the computational effort. To reduce this space two strategies were adopted. The first was to build a table containing all available rules, therefore avoiding search rules not present in the data.

The second was to adopt GbCLONALG in order to reduce the number of clones and to search just interesting areas. The CAISFLO algorithm was tested in a maneuver learning scenario, where a user parked a virtual car and, using the stored data, the respective fuzzy system was achieved, tested and compared with another method. The results proved the efficiency of the algorithm.

References

[1] R. Bishop, “Intelligent Vehicle Applications Worldwide”, *IEEE Inteligent Transportation Systems*, (2000) 78-81.

[2] S. Huang, Wei-Ren, “Safety, Comfort and Optimal Tracking Control in AHS Applications”, *IEEE Control Systems* (1998) 50-64.

[3] J. Baber, Julian Kolodko, Tony Noel, Michael Parent, Ljubo Vlacic, “Cooperative Autonomous Driving”, *IEEE Robotics and Automation Magazine* (2005) 44 - 49.

[4] J. Misener and S. Shladover, “PATH investigation in Vehicle-Roadside Cooperation and Safety: A Foundation for Safety and Vehicle-Infrastructure Integration Research,” *IEEE Proc. Intell. Transp. Syst.,* pp. 9 – 16, 2006.

[5] Qi Chen, et al., “Ohio State University at the 2004 Darpa Grand Challenge: Developing a Completely Autonomous Vehicle”, *IEEE Intelligent Systems* (2004) 8-11.

[6] R. Bishop, “Intelligent vehicle technology and trends”. *Artech House*, 2005.

[7] U. Ozguner, C. Stiller and K. Redmill, “Systems for Safety and Autonomous Behavior in Cars: The DARPA Grand Challenge Experience,” *Proc. IEEE,* vol. 95, no. 2, pp. 397 – 412, Feb. 2007.M.

[8] Kevin B. Korb, Ann E. Nicholson, “Bayesian Artificial Intelligence”, Chapman & Hall/CRC (2004).

[9] Elena Garcia, Maria Antonia Jimenez, Pablo Santos, Manuel Armada, “The Evolution of Robotics Research”, *IEEE Robotics and Automation Magazine* (2007) 90 - 103.

[10] S. Thrun, D.Fox, F. Dellaert, and W. Burgard. “Particle filters for mobile robot localization,” A. Doucet, N. de Freitas and N. Gordon eds. in *Sequential Monte Carlo Methods in Pratice*. New York Springer Verlag, New York, 2001

[11] Ashok K. Goel, Khaled S. Ali, Michael W. Donnellan, Andres Gomes de Silva Garza, Todd J. Callantine, “Multistrategy Adaptive Path Planning”, *IEE Expert* (1994) 57-65.

[12] A. Fujimori, Peter N. Nikiforuk, Madan M. Gupta, “Adaptative Navigation of Mobile Robots with Obstacle Avoidance”, *IEEE Transactions on Robotics and Automation* (1997) Vol.13, 596-602.

[13] R. B. Tilove, “Local obstacle avoidance for mobile robots based on the method of artificial potentials,” in Proc. *IEEE Conf. Robotics Automat., Cincinnati*, OH, 1990, pp. 566–571.

[14] B. H. Krogh, “A generalized potential field approach to obstacle avoidance control,” *in Proc. Int. Robot. Res. Conf., Bethlehem*, PA, Aug. 1984.

[15] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *in Proc. IEEE Conf. Robot. Automat.*, 1985, pp. 500–505.

[16] B. H. Krogh and D. Feng, “Dynamic generation of subgoals for autonomous mobile robots using local feedback information,” *IEEE Trans. Automat. Contr*., vol. 34, pp. 483–493, May 1989.

[17] J.-O. Kilm and P. K. Khosla, “Real-time obstacle avoidance using harmonic potential functions,” *IEEE Trans. Robot. Automat*., vol. 8, pp. 338–349, June 1992.

[18] J. Yen, N. Pfluger, “A Fuzzy Logic Based Extension to Payton and Rosenblatt’s Command Fusion Method for Mobile Robot Navigation”, *IEEE Systems, Man, and Cybernetics* (1995) vol.25, 971 – 978.

[19] S. Leong Tan, J. Gu, “Investigation of Trajectory Tracking Control Algorithms for Autonomouns Mobile Platforms: Theory and Simulation”, *IEEE International Conference on Mechatronics & Automation* (2005) 934-939.

[20] Q. Li, N. Zheng and H. Cheng, “Springrobot: A Prototype Autonomous Vehicle and Its Algorithms for Lane Detection,” *IEEE Trans. Intell. Transp. Syst*., vol. 5, no. 4, pp. 300 – 308, Dec. 2004.

[21] J. C. McCall and M. M. Trivedi, “Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation”, *IEEE Trans. on Intell. Transp. Systems* (2006) vol. 7, 20-37.

[22] A. Kutami, Y.Maruya, H.Takahashi, A.Okuno, “Visual Navigation of Autonomous On-Road Vehicle”, *IEEE International Workshop on Intelligent Robots and Systems*, IROS ’90, (1990)175-180.

[23] Sadayuki Tsugawa, “Vision-Based Vehicles in Japan: Machine Vision Systems and Driving Control Systems”, *IEEE Transactions on Industrial Electronics* (1994) vol. 41, 398-730.

[24] Chris Roman and Charles Reinholtz, Virginia Tech, “Robust Course-Boundary Extraction Algorithms for Autonomous Vehicles”, *IEEE Vision – Based Driving Assistance*, (1998) 32-39.

[25] Yasushi Yagi, Youhimitsu Nishizawa, Masahiko Yachida, "Map-Based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS", *IEEE Transactions on Robotics and Automation*, (1995) vol. 11, 634-647.

[26] C. Thorpe, M. Hebert, T. Kanade and S. Shafer, "Vision and Navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Patt. Analysis Mach. Intell.*, vol. 10, no. 3, pp. 362 – 373, Dec. 1988.

[27] N. Kehtarnavaz, N. Griswold, K. Miller, P. Lescoe, "A Transportable Neural-Network Approach to Autonomous Vehicle Following", *IEEE Transaction on Vehicular Technology* (1998), vol. 47, 694-702.

[28] Stefan K. Gehrig, Fridtjof J. Stein, "Collision Avoidance for Vehicle-Following Systems", *IEEE Transactions on Intelligent Transportation Systems* (2007), vol. 8, 233-244.

[29] I. E. Paromtchik, C. Laugier, "Motion Generation and Control for Parking an Autonomous Vehicle", *Proceeding of the IEEE International Conference on Robotics and Automation*, (1996) 3117-3122.

[30] Ti-Chung Lee, Chi-Yi Tsai, Kai-Tai, "Fast Parking Control of Mobile Robots: A Motion Planning Approach With Experimental Validation", *IEEE Transactions on Control Systems Technology* (2007), vol. 12, 661-676.

[31] Tzuu-Hseng S. Li, Shih-Jie Chang, "Autonomous Fuzzy Parking Control a Car-Like Mobile Robot", *IEEE Transactions on Systems, Man, and Cybernetics* (2003) vol. 33, 451-465.

[32] Yanan Zhao, Emmanuel G. Collins Jr, "Robust Automatic Parallel Parking in Tight Spaces Via Fuzzy Logic", *Robotics and Autonomous Systems* 51 (2005), 111-127.

[33] Hitoshi Miyata, Makoto Ohki, Yasuyuki Yokouchi, Masaaki Ohkita, "Control of the Autonomous Mobile Robot DREAM -1 for a Parallel Parking", *Mathematics and Computers in Simulation* 41(1996) 129-138.

[34] www.isd.mel.nist.gov/projects/rcslib visited January, 2007.

[35] Davis R. S. Punnekkat, N. Audsley, A. Burns, "Flexible scheduling for adaptable real time systems", *IEEE Real Time Technology and Applications Symposium* (1995) 230-239

[36] Rong-wen Huang, Hslan-wei Huang, "Development of Real Time Monitoring System Using Controller Area Network", *Materials Science Forums* (2006) pp. 475-480

[37] Davis RI, Burns A, Bril RJ, Lukkien JJ, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised", *Real-Time Systems* (2007), 239-272

[38] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 155 – 161, Sept. 2002.

[39] J. Naranjo, C. Gonzalez, R. Garcia, T. Pedro and M. Sotelo, "Using Fuzzy Logic in Automated Vehicle Control," *IEEE Intell. Syst.*, vol. 22, no. 1, pp. 36 – 45, Oct. 2006.

[40] S. Shladover, "Path at 20 – History and Major Milestones," *in Proc. IEEE Intell. Transp. Syst.*, pp. 22 – 29, 2006

[41] M. Turk, D. Morgenthaler, K. Gremban, M. Marra, "VITS – A vision system for autonomous land vehicle navigation," *IEEE Trans. Patt. Analysis Mach. Intell.*, vol. 10, no. 3, pp. 342 – 361, May 1988.

[42] S. Kenue, "Lanelok: Detection of Land Boundaries and Vehicle Tracking Using Image-Processing Techniques, Part I: Hough-Transform, Region-Tracing, and Correlation Algorithms," *Proc. Mobile Robots IV*, Soc. of Photooptical Instrumentation Engineers, Bellingham, Wash., 1989., pp. 221-233.

[43] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold and C. Wohler, "Autonomous Driving Goes Downtown," *IEEE Intell. Syst. Applic.*, vol. 13, no. 6, pp. 40 – 48, Nov. – Dec. 1998.

[44] M. Bertozzi, A. Broggi, A. Fascioli and S. Nichele, "Stereo vision-based vehicle detection," *IEEE Proc. Intell. Vehicle Symp.,* Oct. 2000, pp. 39-44.

[45] C. Laugier, I. Paromtchik and M. Parent, "Developing autonomous maneuvering capabilities for future cars," *in IEEE Proc. Intell. Transp. Syst.,* Oct. 1999, pp 68 – 73.

[46] F. Wang, P. Mirchandani and Z. Wang, "The VISTA Project and ITS Applications," *IEEE Intell. Syst.,* vol. 17, no. 6, pp. 72 – 75, Nov. – Dec. 2002.

[47] Thrun S, Montemerlo M, Dahlkamp H, et al. "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol 2, no. 9, pp. 661-692, Sept. 2006.

[48] C. Hatipoglu, U. Ozguner and K. A. Redmill, "Automated Lane Change Controller Design," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 1, pp. 13 – 22, March 2003.

[49] R. Gregor, M. Lutzeler, M. Pellkofer, K. Siedersberger and E. Dickmanns, "EMS-Vision: A Perceptual System for Autonomous Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 48 – 59, Mar. 2002.

[50] W. Li and H. Leung, "Simultaneous registration and fusion of Multiple Dissimilar Sensors for Cooperative Driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 84 – 98, Jun. 2004.

[51] D. Huang and H. Leung, "EM-IMM Based Land-Vehicle Navigation with GPS/INS," *IEEE Intell. Transp. Syst. Conference*, Washington, USA, Oct. 2006.

[52] R. Rogers, "Integrated INU/DGPS for Autonomous Vehicle Navigation," *in Posit. IEEE Local. Navig. Symp.,* Apr. 1996, pp. 471 – 476.

[53] N. El-Sheimy, K. Chiang and A. Noureldin, "The Utilization of Artificial Neural Networks for Multisensor System Integration in Navigation and Positioning Instruments," *IEEE Trans. Intell. Transp. Syst*., vol. 55, no. 5, pp. 1606 – 1615, Oct. 2006.

[54] Stampfle, D. Holz and J. Becker, "Performance Evaluation of Automotive Sensor Data Fusion," *IEEE Proc. Intell. Trans. Syst.,* Sept. 2005, pp. 50-55.

[55] F. Cao, D. Yang, A. Xu, J. Ma, W. Xiao, C. Law, K. Ling and H. Chua, "Low cost SINS/GPS integration for land vehicle navigation," *IEEE Intell. Transp. Syst.,* pp. 910 – 913, 2002.

[56] M. Rydstrom, A. Urruela, E. Strom and A. Svensson, "Low Complexity Tracking Ad-hoc Automotive Sensor Networks," *on IEEE Conf. Sensor and Ad Hoc Communic. Net.,* Oct. 2004, pp. 585 - 591

[57] D. Bevly, J. Ryu and J. Gerdes, "Integrating INS Sensors with GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tir Cornering Stiffness," *IEEE Trans. Intell. Transp. Syst*., vol. 7, no. 4, pp. 483 – 493, Dec. 2006.

[58] L. Bento, U. Nunes, F. Moita and A. Surrecio, "Sensor Fusion for Precise Autonomous Vehicle Navigation in Outdoor Semi-structured Enviroments," *in Proc. Of the 8th International IEEE Conf. on Intell. Transp. Syst.*, Vienna, Austria, Sept. 2005, pp. 245–250.

[59] J. Naranjo, C. Gonzalez, T. Pedro, R. Garcia, J. Alonso and M. Sotelo, "AUTOPIA Architecture for Automatic Driving and Maneuvering," *IEEE Proc. Intell. Transp. Syst.,* 2006, pp. 1220-1225.

[60] M. Russell, C. Drubin, A. Marinilli, W. Woodington and M. Checcolo, "Integrated Automotive Sensors," *IEEE Trans. Micr. Theory Tech.,* vol. 50, no. 3, pp. 674 – 677, March 2002.

[61] J. Blum, A. Eskandarian, "The Threat of Intelligent Collisions," *IEEE IT Professional,* vol. 6, no. 1, pp. 24 -29, Jan. - Feb. 2004.

[62] B. Kamali, "Some applications of wireless communications in intelligent vehicle highway systems," *IEEE Aerosp. Electron. Syst. Mag.,* vol. 11, no. 11, pp. 8 – 12, Nov. 1996.

[63] A. M. Kirson, "RF Data Communications Considerations in Advanced Driver Information Systems," *IEEE Trans. Vehic. Tech*., vol. 40, no. 1, pp. 51 – 55, Feb. 1991.

[64] S. Tsugawa, "Inter-Vehicle Communications and their Applications to Intelligent Vehicle: an Overview," *IEEE Intell. Vehicle Symp.,* vol. 2 pp. 17 – 21, June 2002.

[65] L. Yang and F. Wang, "Driving into Intelligent Spaces with pervasive Communications," *IEEE Intell. Syst.,* vol. 22, no. 1, pp. 12 – 15, Jan. – Feb. 2007.

[66] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang and D. McMahon, "Automated Vehicle Control Developments in the PATH Program," *IEEE Trans. Vehicular Tech.,* vol. 40, no. 1, pp. 114 – 130, Feb. 1991.

[67] Nga-Viet Nguyen, et al., "An Observation Model Based on Polyline Map for Autonomous Vehicle Localization", *IEEE* (2006) 2427-2431.

[68] Nga-Viet Nguyen, et al., "A New Observation Model to Solve Vehicle Localization Problem", *IEEE*, (2006)

[69] Ranka Kulic and Zoran Vukic, "Behavior Cloning and Obstacle Avoiding for Two Different Autonomous Vehicles", *IEEE* (2006) 1227-1232.

[70] Ranka Kulic and Zoran Vukic, "Autonomous Vehicle Obstacle Avoiding and Goal Position Reaching by Behavioral Cloning", *IEEE* (2006) 3939-3944.

[71] Zhiyu Xiang and Umit Ozgilmer, "A 3D Positioning System for Off-Road Autonomous Vehicles", *IEEE* (2005) 130-135.

[72] J. Illingworth and J. Kittler, "A Survey of the Hough Transform", *CVGIP*, vol. 44, pp. 87-116, 1988.

[73] Greg Welch and Gary Bishop, "An Introduction to Kalman Filters", *Course at the SIGGRAPH*, (2001)

[74] Frank Dellaert, et al., "Model Based Car Tracking Integrated with a Road Follower", *Proceedings of the IEEE International Conference on Robotics and Automation*, (1998) 1889-1894.

[75] Nasser Kehtarnavaz, et al., "Visual Control of an Autonomous Vehicle (BART) – The Vehicle Following Problem", *IEEE Transactions on Vehicular Technology* (1991) 654-662.

[76] M. Kobayashi, et al., "A Method of Visual Servoing for Autonomous Vehicles", *IEEE* (1996) 371-376.

[77] Lyle N. Long et al., "A Review of Intelligent Systems Software for Autonomous Vehicles", *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Application*, (2007) 69-76.

[78] Pal, N. R., Chakraborty, S.: "Fuzzy Rule Extraction from ID3-Type Decision Trees for Real Data". *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics 05*, vol. 31, pp. 745-754 (2001)

[79] Quinlan, J. R.,"C4.5.:Programs for Machine Learning". San Mateo, Inc. MorganKaufmann (1993)

[80] Alves, R. T., Delgado, R.M., Lopes, H.S., Freitas, A.A.: "An Artificial Immune System for Fuzzy-Rule Induction in Data Mining". In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervós, J.J. (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 1011-1020 Springer, Heidelberg (2004)

[81] Wang, L., Mendel, J.M.: “Generating Fuzzy Rules by Learning from Examples.” *IEEE Transactions on Systems, Man, and Cybernetics* 6, vol. 22, pp. 1414-1427 (1992)

[82] Abido, M.A.: “A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch”, *Int. Journal of Electrical Power and Energy Systems* 2, vol. 25, pp. 97-105(2003)

[83] Li, Y., Ha, M., Wang, X.: “Principle and Design of Fuzzy Controller Based on Fuzzy Learning from Examples”, In: *Proc. of the 1nd Int. Con. on Machine Learning and Cybernetics* 3, 1441-1446 (2002)

[84] Pei, Z.: “A Formalism to Extract Fuzzy If-Then Rules from Numerical Data Using Genetic Algorithms”, In: *Int. Symposium on Evolving Fuzzy Systems*, pp. 143-147 (2006)

[85] Abe, S., Lan, M.: “Fuzzy Rules Extraction Directly from Numerical Data for Function Approximation” *IEEE Transactions on Systems, Man, and Cybernetics* 01, vol. 25, pp. 119-129 (1995)

[86] Zhao, Y. Collins, E.G., Dunlap, D.: “Design of genetic fuzzy parallel parking control systems”, In: *Proc. American Control Conference,* vol. 5, 4107-4112 (2003)

[87] Castro, L.N., Zubben, F.J.V.: “Learning and Optimization Using the Clonal Selection Principle”, *IEEE Transactions on Evolutionary Computation* 3, vol. 6, 239-251 (2002)

[88] Honorio, L.M., Leite da Silva, A.M. e Barbosa, D.A.: “A Gradient-Based Artificial Immune System Applied to Optimal Power Flow Problems”, In: Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) *Artificial Immune Systems ICARIS 2007*. LNCS, vol. 4628, pp. 1-12, Springer, Heidelberg (2007)

[89] Honorio, L M.: “Virtual Manufacture Software.” Information Technology Institute/UNIFEI. Itajubá, Brazil (2005), [Online]. Available: www.virtualmanufacturing.unifei.edu.br

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-236

Paraconsistent Autonomous Mobile Robot Emmy III

Jair Minoro ABE [a], Cláudio Rodrigo TORRES [b,c] Germano LAMBERT-TORRES [b], João Inácio DA SILVA FILHO [d] and Helga Gonzaga MARTINS [b]
[a] *Paulista University, UNIP - Dr. Bacelar, 1.212, São Paulo, SP - Brazil*
[b] *Itajuba Federal University - Artificial Intelligence Application Group, Av. BPS 1303, Itajuba, MG - Brazil*
[c] *Universidade Metodista de São Paulo, São Bernardo do Campo, SP - Brazil*
[d] *Universidade Santa Cecília – UNISANTA, Santos, SP - Brazil*

Abstract. This work presents some improvements regarding to the autonomous mobile robot Emmy based on Paraconsistent Annotated Evidential Logic Eτ. A discussion on navigation system is presented.

Keywords. Automation, paraconsistent logic, robotics, navigation system, logic controller

Introduction

It is well known the use of non-classical logics in automation and robotics. In real applications, classical logic is inadequate for several reasons. The main point is that all concepts of real world encompass some imprecision degree. In order to overcome these limitations, several alternative systems were proposed. Maybe the most successful non-classical system is the so-called Fuzzy set theory [16]. In this work we employ another promising non-classical logic, namely the paraconsistent annotated systems. They've inspired applications in a variety of themes. Particularly in robotics, it was built some interesting autonomous mobile robots that can manipulate imprecise, inconsistent and paracomplete data. One of the robot series dubbed Emmy[1)], based on a particular annotated system, namely, the paraconsistent annotated evidential logic Eτ [1], began with the 1st prototype studied in [2], [3]. Subsequently, some improvements were made in its 2nd prototype Emmy II [4] and in this paper we sketch the 3rd prototype discussing a navigation system.

1. Paraconsistent, paracomplete, and non-alethic logics

In what follows, we sketch the non-classical logics discussed in the paper, establishing some conventions and definitions.

[1)] The name Emmy is in homage to the mathematician Emmy Nöether (1882-1935). Such name was proposed by N.C.A. da Costa and communicated to J.M. Abe in 1999, University of Sao Paulo.

Let T be a theory whose underlying logic is L. T is called inconsistent when it contains theorems of the form A and $\neg A$ (the negation of A). If T is not inconsistent, it is called *consistent.* T is said to be *trivial* if all formulas of the language of T are also theorems of T. Otherwise, T is called *non-trivial.* When L is classical logic (or one of several others, such as intuitionistic logic), T is inconsistent if T is trivial. So, in trivial theories the extensions of the concepts of formula and theorem coincide. Paraconsistent *logic* is a logic that can be used as the basis for inconsistent but non-trivial theories. A *theory* is called *paraconsistent* if its underlying logic is a paraconsistent logic. Issues such as those described above have been appreciated by many logicians. In 1910, the Russian logician Nikolaj A. Vasil'év (1880-1940) and the Polish logician Jan Łukasiewicz (1878-1956) independently glimpsed the possibility of developing such logics. Nevertheless, Stanislaw Jaśkowski (1996-1965) was in 1948 effectively the first logician to develop a paraconsistent system, at the propositional level [9]. His system is known as 'discussive propositional calculus'. Independently, some years later, the Brazilian logician Newton C.A. da Costa (1929-) constructed for the first time hierarchies of paraconsistent propositional calculi C_i, $1 \leq i \leq \omega$ of paraconsistent first-order predicate calculi (with and without equality), of paraconsistent description calculi, and paraconsistent higher-order logics (systems NF_i, $1 \leq i \leq \omega$). Also, independently of Da Costa [10], Nels David Nelson (1918-2003) [11] has considered a paraconsistent logic as a version of his known as constructive logics with strong negation.

Nowadays, paraconsistent logic has established a distinctive position in a variety of fields of knowledge.

Another important class of non-classical logics are the paracomplete logics. A logical system is called *paracomplete* if it can function as the underlying logic of theories in which there are formulas such that these formulas and their negations are simultaneously false. Intuitionistic logic and several systems of many-valued logics are paracomplete in this sense (and the dual of intuitionistic logic, Brouwerian logic, is therefore paraconsistent).

As a consequence, paraconsistent theories do not satisfy the principle of non-contradiction, which can be stated as follows: of two contradictory propositions, i.e., one of which is the negation of the other, one must be false. And, paracomplete theories do not satisfy the principle of the excluded middle, formulated in the following form: of two contradictory propositions, one must be true.

Finally, logics which are simultaneously paraconsistent and paracomplete are called *non-alethic logics.*

2. Paraconsistent Annotated Evidential Logic Eτ

Annotated logics are a family of non-classical logics initially used in logic programming by [12]. An extensive study of annotated logics was made in [1]. Some applications are summarized in [13]. In view of the applicability of annotated logics to differing formalisms in computer science, it has become essential to study these logics more carefully, mainly from the foundational point of view.

In general, annotated logics are a kind of paraconsistent, paracomplete, and non-alethic logic. The latter systems are among the most original and imaginative systems of non-classical logic developed in the past century.

The atomic formulas of the logic $E\tau$ are of the type $p_{(\mu,\ \lambda)}$, where $(\mu, \lambda) \in [0, 1]^2$ and [0, 1] is the real unitary interval (p denotes a propositional variable). $p_{(\mu,\ \lambda)}$ can be intuitively read: "It is assumed that p's favorable evidence is μ and contrary evidence is λ." Thus, $p_{(1.0,\ 0.0)}$ can be read as a true proposition, $p_{(0.0,\ 1.0)}$ as false, $p_{(1.0,\ 1.0)}$ as inconsistent, $p_{(0.0,\ 0.0)}$ as paracomplete, and $p_{(0.5,\ 0.5)}$ as an indefinite proposition.

Also we introduce: Uncertainty Degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$; Certainty Degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$ $(0 \leq \mu, \lambda \leq 1)$; an order relation is defined on [0, 1]2: $(\mu1, \lambda1) \leq (\mu2, \lambda2) \Leftrightarrow \mu1 \leq \mu2$ and $\lambda1 \leq \lambda2$, constituting a lattice that will be symbolized by τ. With the uncertainty and certainty degrees we can get the following 12 output states: extreme state and non-extreme states, showed in the Table 1.

Table 1. Extreme and Non-extreme states

Extreme States	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	QV→T
False	F	Quasi-true tending to Paracomplete	QV→⊥
Inconsistent	T	Quasi-false tending to Inconsistent	QF→T
Paracomplete	⊥	Quasi-false tending to Paracomplete	QF→⊥
		Quasi-inconsistent tending to True	QT→V
		Quasi-inconsistent tending to False	QT→F
		Quasi-paracomplete tending to True	Q⊥→V
		Quasi-paracomplete tending to False	Q⊥→F

All states are represented in the next figure.

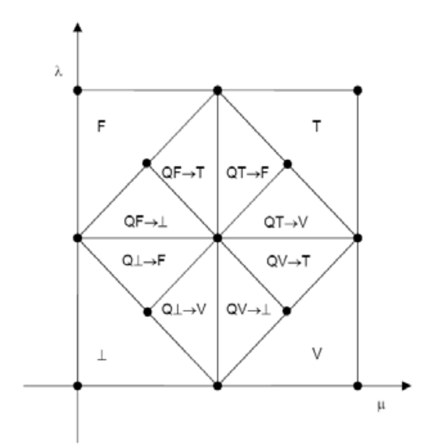


Figure 1. Lattice τ

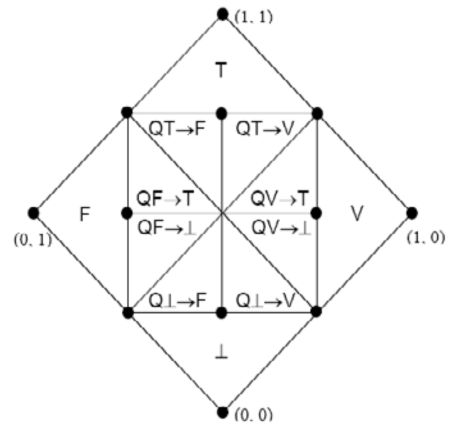


Figure 2. Lattice τ - Hasse diagram

Some additional control values are:

V_{cic} = maximum value of uncertainty control = C_3
V_{cve} = maximum value of certainty control = C_1
V_{cpa} = minimum value of uncertainty control = C_4
V_{cfa} = minimum value of certainty control = C_2

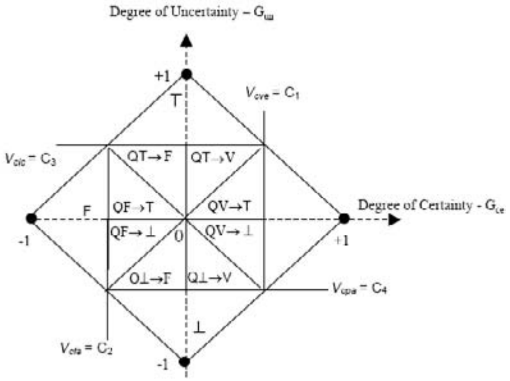


Figure 3. Certainty and Uncertainty degrees

3. Paracontrol - Logical Controller

The Paracontrol [20] is an electronic materialization of the Para-analyzer algorithm [2], [15], which is basically an electronic circuitry, which treats logical signals in a context of logic Eτ. Such circuitry compares logical values and determines domains of a state lattice corresponding to output value. Favorable evidence and contrary evidence degrees are represented by voltage. Certainty and Uncertainty degrees are determined by analyze of operational amplifiers. The Paracontrol comprises both analogical and digital systems and it can be externally adjusted by applying positive and negative voltages. The Paracontrol was tested in real-life experiments with an autonomous mobile robot Emmy, whose favorable/contrary evidences coincide with the values of ultrasonic sensors and distances are represented by continuous values of voltage. The figure 4 shows the circuit of the Paracontrol.

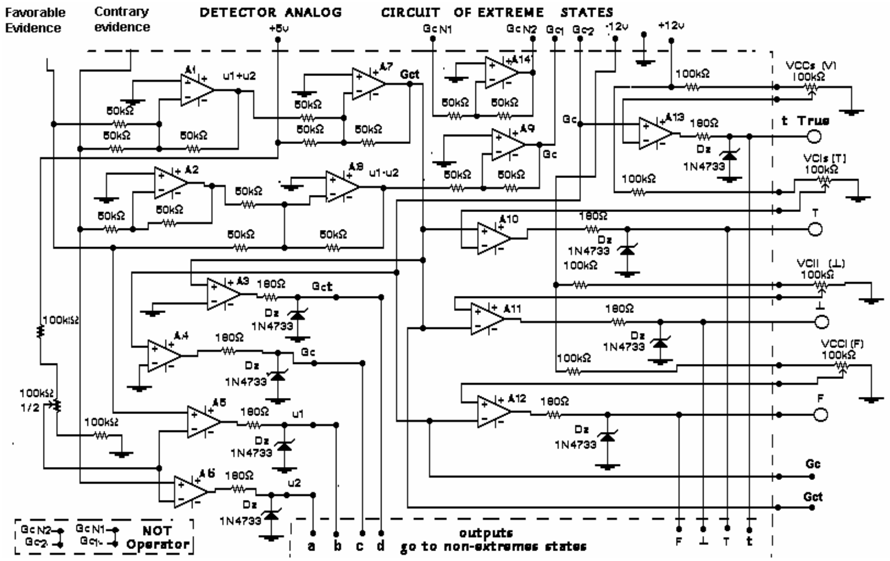


Figure 4. Paracontrol circuitry

4. The Autonomous Mobile Robot Emmy

The controller Paracontrol was applied in this series of autonomous mobile robots. In some previous works [2], [19] is presented the autonomous mobile robot Emmy. The figure 5 shows the autonomous mobile robot Emmy. The Emmy robot consists of a circular mobile platform of aluminum 30 cm in diameter and 60 cm height. While moving in a non-structured environment the robot Emmy gets information about presence/absence of obstacles using the sonar system called Parasonic [3].

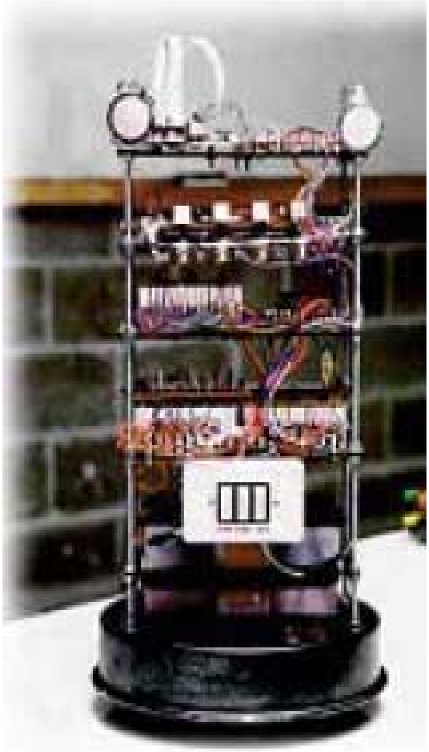


Figure 5. The autonomous mobile robot Emmy

Parasonic is an electronic circuitry that Emmy robot uses to detect obstacles in its path. The Parasonic transforms the distances to the obstacle into electric signals of the continuous voltage ranging from 0 to 5 volts. The Parasonic is basically composed of two ultrasonic sensors of type POLAROID 6500 controlled by an 8051 microcontroller. The 8051 is programmed to carry out synchronization between the measurements of the two sensors and the transformation of the distance into electric voltage.

The Parasonic generates the favorable evidence degree value (μ) and the contrary evidence degree value (λ). They are a continuous voltage ranging from 0 to 5 volts. The Paracontrol receives these signals from the Parasonic. The figure 6 shows the basic structure of Emmy robot.

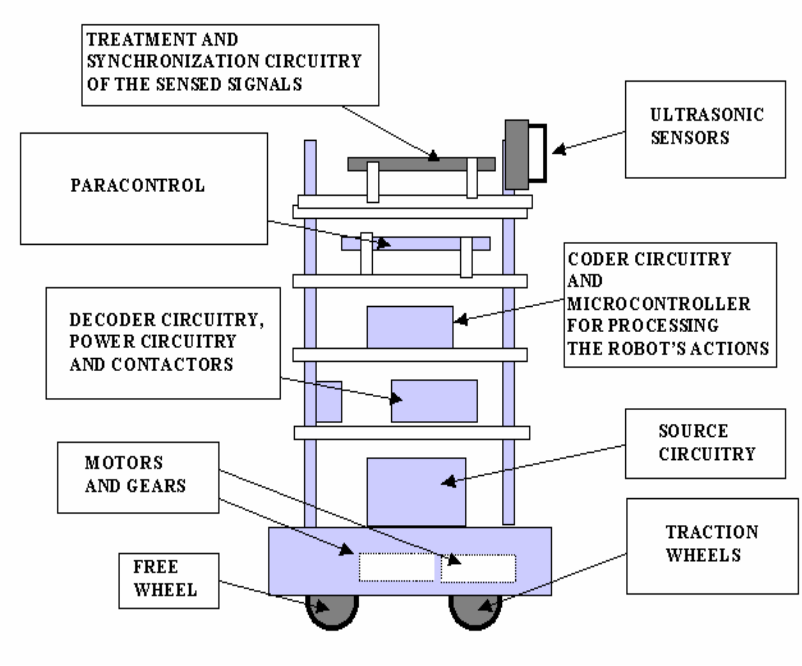

TREATMENT AND SYNCHRONIZATION CIRCUITRY OF THE SENSED SIGNALS
ULTRASONIC SENSORS
PARACONTROL
CODER CIRCUITRY AND MICROCONTROLLER FOR PROCESSING THE ROBOT'S ACTIONS
DECODER CIRCUITRY, POWER CIRCUITRY AND CONTACTORS
SOURCE CIRCUITRY
MOTORS AND GEARS
TRACTION WHEELS
FREE WHEEL

Figure 6. Basic structure of Emmy robot

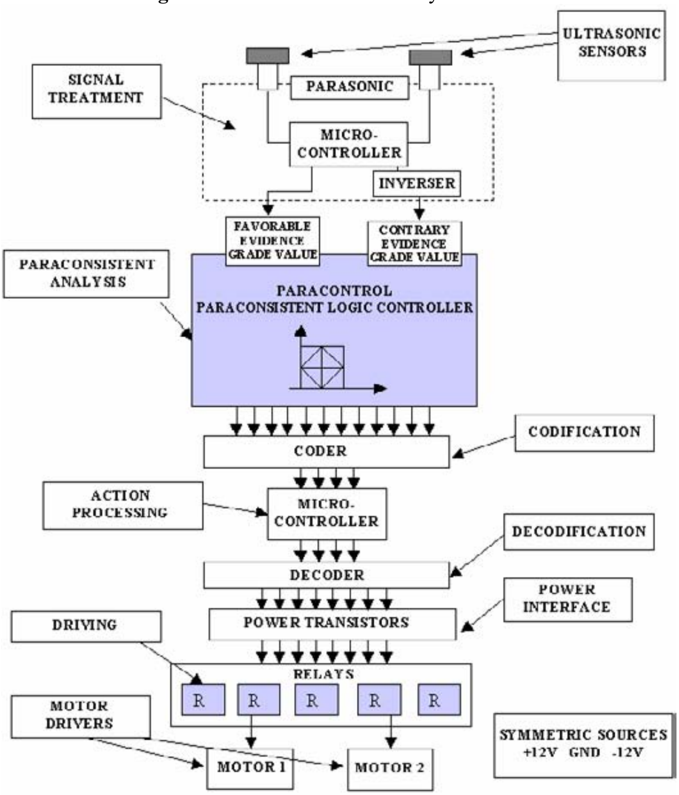


Figure 7. Main components of Emmy robot

In the figure 7 may be seen the main components of Emmy robot.

The description of the Emmy robot components is the following.

- Ultrasonic sensors: two ultrasonic sensors are responsible for emitting ultrasonic waves and for detecting the return of them.
- Signal treatment: in the Parasonic there is a microcontroller. It sends to the sensors a signal that makes them to emit ultrasonic waves. When the ultrasonic waves return, the sensors send to the microcontroller a signal. The microcontroller measures the time lasted between the sending and the returning of the ultrasonic waves. So, the microcontroller is able to determine the distance between the sensors and the obstacles in front of them. The Paracontrol generates a continuous voltage ranging from 0 to 5 volts proportional to the distance between the sensor and the obstacle. This signal is considered the favorable evidence degree value on the proposition "The front of the robot is free". In the same way the Paracontrol generates a continuous voltage ranging from 5 to 0 volts related to the contrary evidence degree.
- Paraconsistent analysis: the Paracontrol makes the logical analysis of the signals according to the logic $E\tau$.
- Codification: the coding circuitry changes a 12-digit word to a code of 4 digits.
- Action processing: a microcontroler processes the 4 digit code generated by the coding circuitry determining the sequence of relays must be actuated for the robot performs the right movement.
- Decodification: the decoding circuitry changes a 4-digit word to a code of 12 digits.
- Power interface: transistors amplify the signals of the 12-digit word generated by the decoding circuitry. Then the signals can actuate the relays.
- Driving: relays are responsible for actuate the DC motors M1 and M2.
- Motor driver: two DC motors are responsible for moving the robot.
- Sources: two batteries composing a ± 12 volts symmetric source feed the Emmy robot electric circuitries.

Emmy II robot is an improvement of Emmy robot and it is described in what follows.

5. Robot Emmy II

Searching the Paracontrol, the robot Emmy controller, we perceived that the robot movements could be bettered by programming conveniently the no extreme logic state outs. This work shows a robot Emmy Paracontrol modification that allows a better Paracontrol performance.

This new Paracontrol version also is used to control an autonomous mobile robot named as Emmy II.

The platform used to assemble the Emmy II robot measures approximately 23cm height and 25cm of diameter (circular format). The main components of Emmy II are a microcontroller from 8051 family, two ultrasonic sensors, and two DC motors. Figure 8 shows an Emmy II robot simplified block diagram.

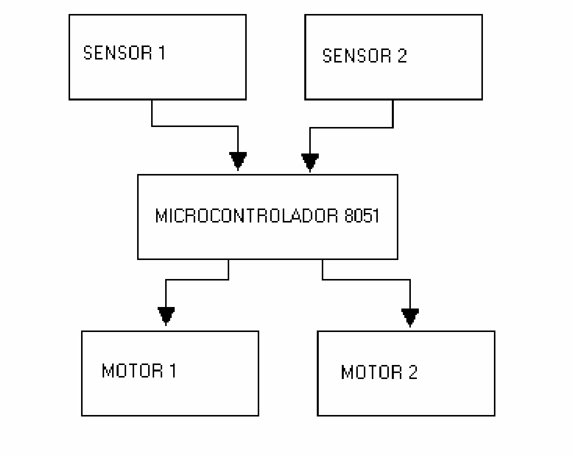


Figure 8. Emmy II robot simplified block diagram.

The ultrasonic sensors are responsible for verifying whether there is any obstacle in front of the robot. The signals generated by the sensors are sent to the microcontroller. These signals are used to determine the favorable evidence degree value (μ) and the contrary evidence degree value (λ) on the proposition "The front of the robot is free".

The Paracontrol, recorded in the internal memory of the microcontroller, uses the evidence degrees in order to determine the robot movements. The microcontroller is also responsible for applying power to the DC motors.

Figure 9 shows the Emmy II mechanical structure.

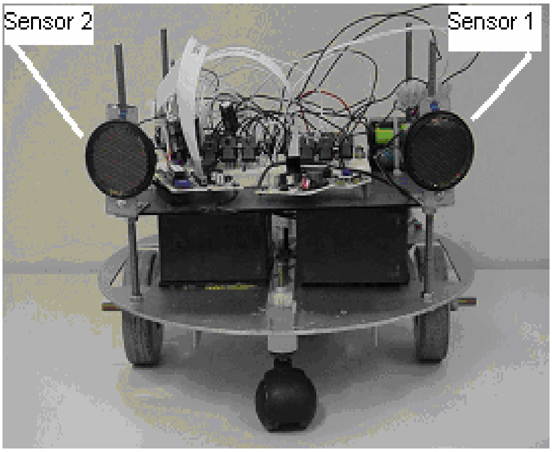


Figure 9. Emmy II Mechanical Structure.

Figure 10 shows the decision state lattice that the Emmy II robot uses to determine the movement to perform.

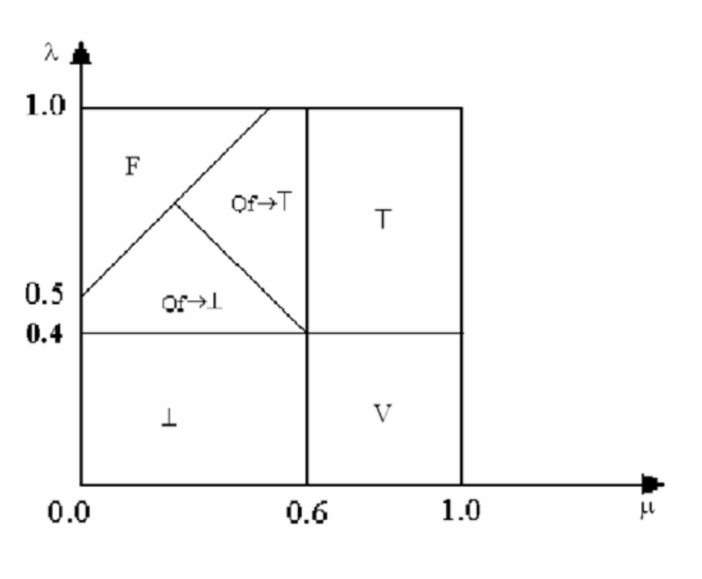


Figure 10. Logical output lattice of Emmy II

Table 2 shows the actions related to each possible logic state. Each robot movement lasts approximately 0,4 seconds.

Table 2. Logical states and action

Symbol	State	Action
V	True	Robot goes ahead
F	False	Robot goes back
⊥	Paracomplete	Robot turns right
T	Inconsistent	Robot turns left
QF→⊥	Quasi-false tending to paracomplete	Robot turns right
QF→T	Quasi-true tending to inconsistent	Robot turns left

5.1 Robot Movement Speed Control

The Paracontrol new version allows the robot movement speed control. The maximum value of certainty control (V_{cve}), minimum value of certainty control (V_{cfa}), maximum value of uncertainty control (V_{cic}) and minimum value of uncertainty control (V_{cpa}) are used to determine the robot movement speed.

Figure 11 shows the perfectly undefined line, or, Gce ≡ 0.

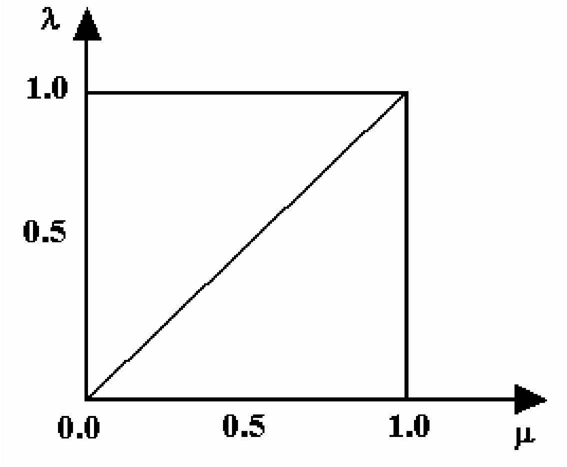


Figure 11. Perfectly undefined line

When Gce(μ, λ) = μ - λ > 0, the certainty degree "moves away" from the perfectly undefined line "toward" the True State (V), as showed in figure 12.

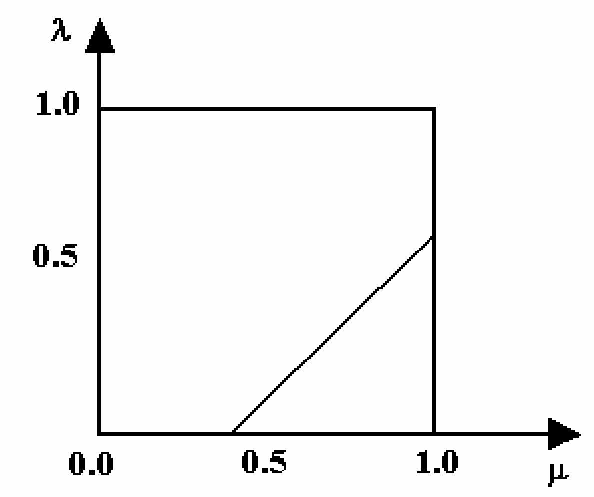


Figure 12. Line for Gce(μ, λ) = k, 0 < k < 1

Thus, the movement speed varies proportionally to the maximum value of certainty control (Vcve), being minimum when Gce assumes values from the perfectly undefined line and maximum when Gce assumes value equal to 1.

At False, Quasi-false tending to Inconsistent and Quasi-false tending to Paracomplete states, or, when Gce(μ, λ) = μ - λ < 0, the Certainty Degree "moves away" from the perfectly undefined line "toward" the False state, as showed in figure 13.

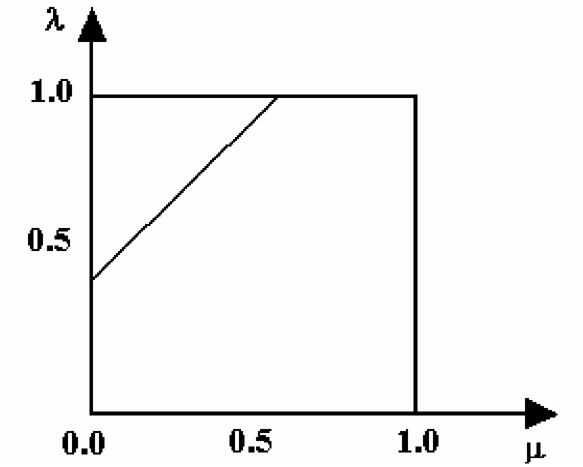


Figure 13. Line for Gce (μ, λ) = k, -1 < k < 0.

So, the movement speed depends on the minimum value of certainty control (Vcfa).

Figure 14 shows the perfectly defined line. It is characterized by the follow equation: Gun(μ, λ) = μ + λ - 1 = 0.

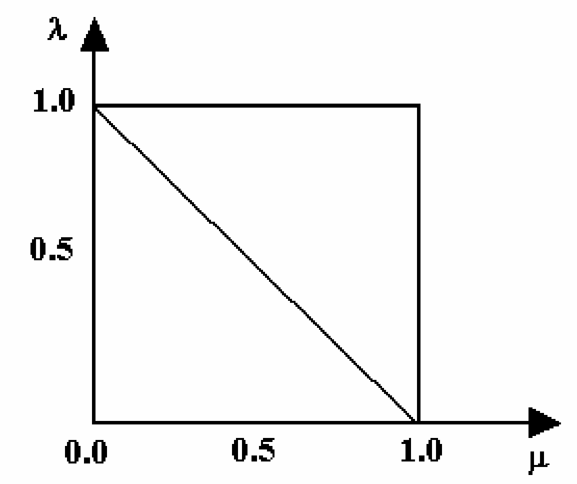


Figure 14. Perfectly defined line

When $G_{un}(\mu, \lambda) = \mu + \lambda - 1 = k$, $0 < k < 1$, the Uncertainty Degree "moves away" from the perfectly defined line "toward" the Inconsistent state, as showed in figure 15.

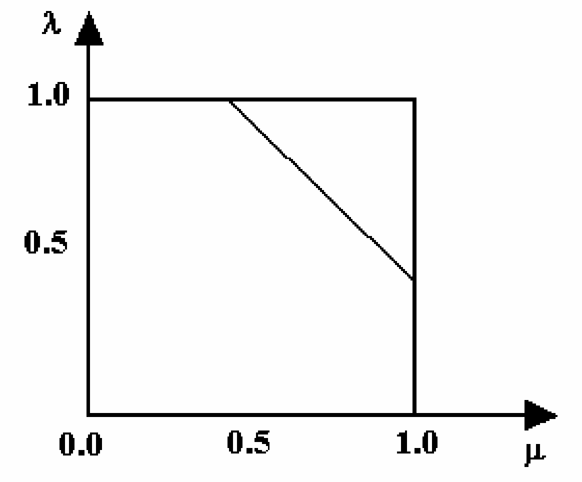


Figure 15. Line for Gun (μ, λ) = k, 0 < k <1

Hence, in this situation, the movement speed depends on maximum value of uncertainty control (G_{cic}), being minimum when Gun assumes values from the perfectly defined line and maximum when Gun assumes value equal to 1.

When $G_{un}(\mu, \lambda) = \mu + \lambda - 1 = k$, $-1 < k < 0$ the Uncertainty Degree "moves away" from the perfectly defined line "toward" the Paracomplete state, as showed in figure 16.

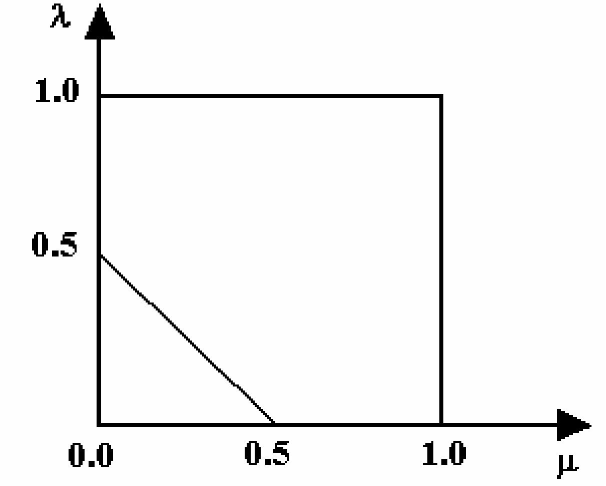


Figure 16. Line for $G_{un} = k$, $-1 < k < 0$.

At this situation, the speed movement depends on the minimum value of uncertainty control (Vcpa).

Summarizing:

- For $\mu > 0.5$ and $\lambda \leq 0.5$
  $G_{ce}(\mu, \lambda)$ or $G_{cve}(\mu, \lambda)$ determines the movement speed.

- For $\mu \leq 0.5$ and $\lambda > 0.5$
  $|G_{ce}(\mu, \lambda)|$ or $|G_{cfa}(\mu, \lambda)|$ determines the movement speed.
- For $\mu \leq 0.5$ and $\lambda \leq 0.5$
  $|G_{un}(\mu, \lambda)|$ or $|G_{cpa}(\mu, \lambda)|$ determines the movement speed.
- For $\mu > 0.5$ and $\lambda > 0.5$
  $G_{in}(\mu, \lambda)$ or $G_{cic}(\mu, \lambda)$ determines the movement speed.

The speed movement control is not implemented in Emmy II robot.
Following, we describe in detail the Emmy II robot.

5.2 Emmy II Robot Description

The robot description is in four stages:

- Source circuitry.
- Sensor circuitry.
- Control circuitry.
- Power circuitry.

5.2.1 – Source circuitry

The source circuitry aim is to supply 5, 12 and 16,8 Volts DC to the others robot Emmy II circuitries.

Figure 17 shows the source circuitry electric scheme.

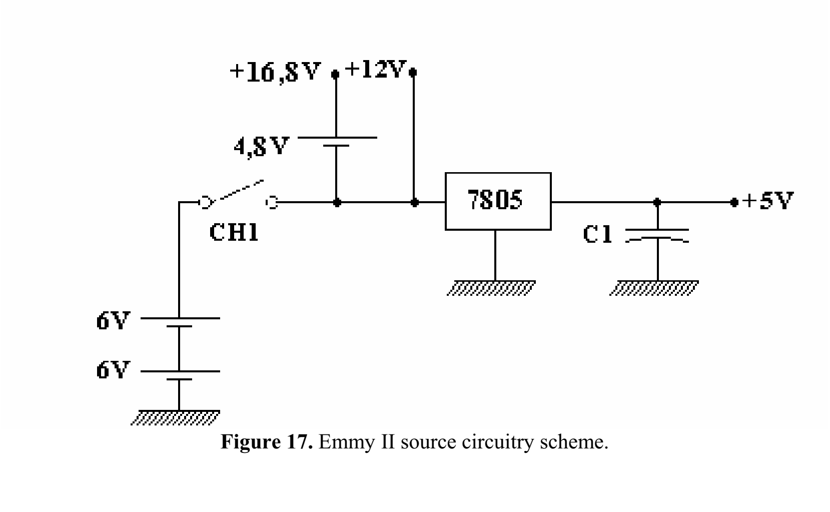


Figure 17. Emmy II source circuitry scheme.

5.2.2 – Sensor circuitry

The "Polaroid 6500 Series Sonar Ranging Module" [5] was used in Emmy II robot. The sonar ranging modules gets the environment information about the place where the robot is in locomotion. This information is sent to the microcontroller.

This device has three inputs (V_{cc}, G_{nd} and Init) and one output (Echo). When INIT is taken high by the microcontroller, the sonar ranging module transmits 16 pulses at 49,4 kilohertz. After receiving the echo pulses, which causes the ECHO output high, the sonar ranging module sends ECHO signal to the microcontroller. Then, with the time interval between INIT sending and ECHO receiving, the microcontroller may determine the distance between the robot and the obstacle. The figure 18 shows the Emmy II sensor circuitry scheme. The microcontroller 89C52 from 8051 family is responsible to control the Emmy II robot.

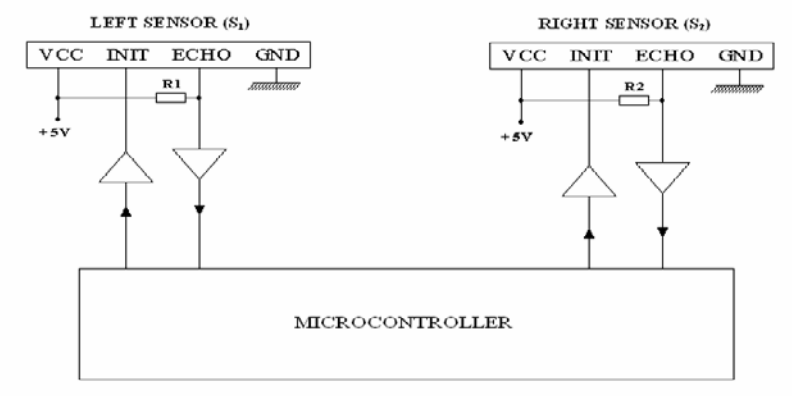


Figure 18. Sensor circuitry used by Emmy II robot.

5.2.3 – Control circuitry

This is the main important Emmy II robot circuitry. It is responsible for determine the distance between the robot and the obstacles, to change these distances to the favorable and contrary evidence degree values, to run the Paracontrol program and to apply power to the DC motors.

The microcontroller 89C52 from 8051 family was chosen to control the Emmy II robot. Its input/output port 1 is used to send and receive signals to and from Sensor Circuitry and Power Circuitry. Buffers make the interface between the microcontroller and the other circuitries.

The microcontroller I/O Port 1 has 8 pins. The function of each pin is the following:

- Pin 0: INIT of sonar ranging module 1 (S_1).
- Pin 1: ECHO of sonar ranging module 1 (S_1).
- Pin 2: INIT of sonar ranging module 2 (S_2).
- Pin 3: ECHO of sonar ranging module 2 (S_2).
- Pin 4: When it is taken high (+5 Volts), DC motor 1 is power supplied.
- Pin 5: When it is taken high (+5 Volts), DC motor 2 is power supplied.

- Pin 6: When it is taken low (0 Volts), DC motor 1 spins around forward while pin 4 is taken high (+5 Volts). When it is taken high (+5 Volts), DC motor 1 spins around backward while pin 4 is taken high (+5 Volts).

- Pin 7: When it is taken low (0 Volts), DC motor 2 spins around forward while pin 5 is taken high (+5 Volts). When it is taken high (+5 Volts), DC motor 2 spins around backward while pin 5 is taken high (+5 Volts).

Figure 19 shows the main microcontroller connections.

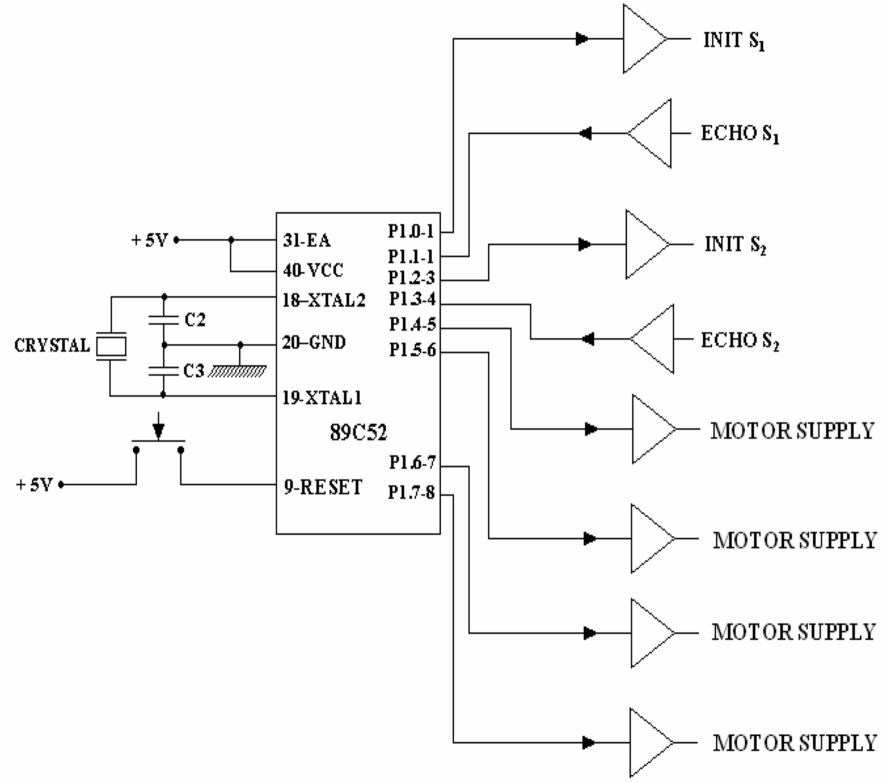


Figure 19. Robot Emmy II main microcontroller connections

5.2.4 – Power circuitry

Two DC motors supplied by 12 Volts DC are responsible for Emmy II robot movements. The Paracontrol, through the microcontroller, determines which DC motor must be supplied and which direction it must spin around. Basically, power field effect transistors – MOSFETs, compose the power interface circuitry.

Figure 20 shows Emmy II robot power interface circuitry.

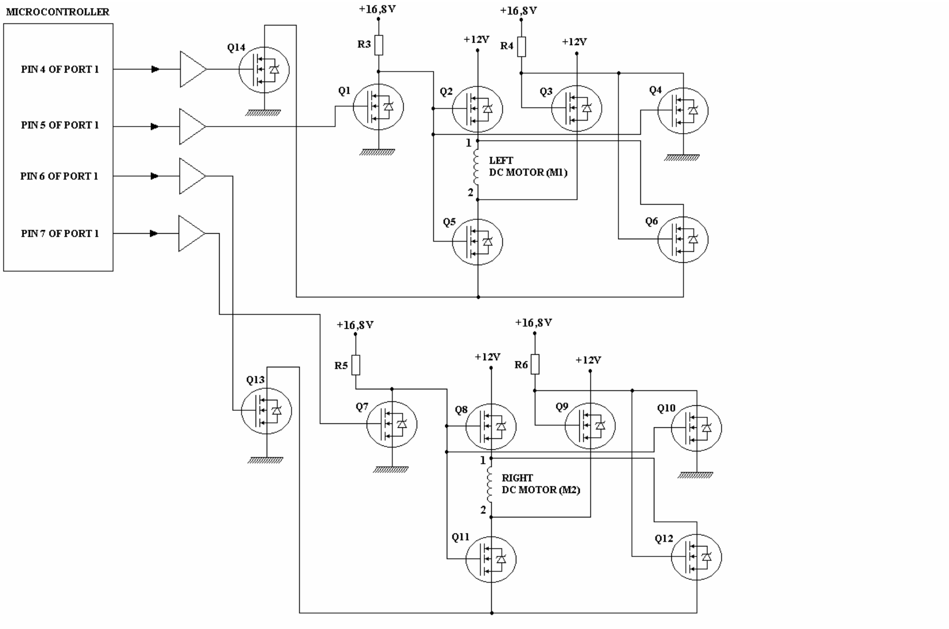


Figure 20. Emmy II robot power interface circuitry

5.3 Emmy II Robot Program

Next, the robot Emmy II program is described.

5.3.1 –Favorable evidence degree determination

The sonar ranging module 1 INIT is connected to microcontroller 89C52 I/O port 1 pin 0. When the microcontroller 89C52 I/O port 1 pin 0 is taken high, the sonar ranging module 1 transmits sonar pulses. The sonar ranging module 1 ECHO is connected to microcontroller 89C52 I/O port 1 pin 1.

So, when this pin is taken high, it means that sonar echo pulses have just returned. Hence it is possible to determine the distance between sonar ranging module 1 and the obstacle in front of it. The robot can measures distances between 0 cm and 126 cm. Therefore, a distance of 0 cm means that the favorable evidence degree value (μ) on the proposition “The front of the robot is free” is 0. And a distance of 126 cm means that the favorable evidence degree value (μ) on the proposition “The front of the robot is free” is 1.

5.3.2 – Contrary evidence degree determination

In the same way as described for the favorable evidence value determination, the contrary evidence degree value (λ) on the proposition "The front of the robot is free" is determined. But now, a distance of 0 cm between the sonar ranging module 2 and the obstacle in front of it means that the contrary evidence degree value (λ) on the proposition "The front of the robot is free" is 1. And a distance of 126 cm between the sonar ranging module 2 and the obstacle in front of it means that the contrary evidence degree value (λ) on the proposition "The front of the robot is free" is 0.

5.4 Emmy II Robot Movements

The decision for each logic state is the following:

- V state: Robot goes ahead. DC motors 1 and 2 are supplied for spinning around forward.
- F state: Robot goes back. DC motors 1 and 2 are supplied for spinning around backward.
- ⊥ state: Robot turns right. Just DC motor 1 is supplied for spinning around forward.
- T state: Robot turns left. Just DC motor 2 is supplied for spinning around forward.
- QF→⊥ state: Robot turns right. Just DC motor 2 is supplied for spinning around backward.
- QF→T state: Robot turns left. Just DC motor 1 is supplied for spinning around backward.

Each robot movement lasts approximately 0,4 seconds.

5.5 Tests

Aiming to verify Emmy II robot functionally, we performed 4 tests. Basically, counting how many collisions there were while the robot moved in an environment similar to the one showed in figure 21 composed the tests.

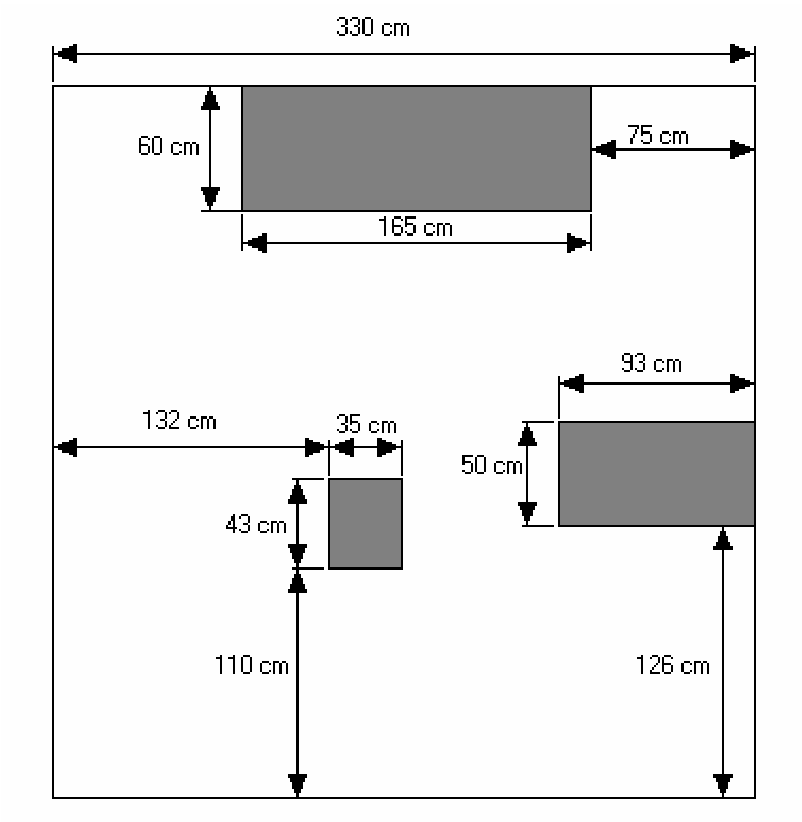


Figure 21. The robot Emmy II test environment.

The time duration and results for each test were the following:

Test 1: Duration: 3 minutes and 50 seconds. Result: 13 collisions.

Test 2: Duration: 3 minutes and 10 seconds. Result: 7 collisions.

Test 3: Duration: 3 minutes and 30 seconds. Result: 10 collisions.

Test 4: Duration: 2 minutes and 45 seconds. Result: 10 collisions.

The sonar ranging modules used in Emmy II robot can't detect obstacles closer than 7,5 cm. The sonar ranging modules transmit sonar pulses and wait for the sonar pulses return (echo) to determine the distance between the sonar ranging modules and the obstacles; nevertheless, sometimes the echo don't return, it reflects to another direction. These are the main causes to the robot collisions. Adding more sonar ranging modules and modifying the Paracontrol may solve it

The robot collision causes are the following:

Test 1: Collisions: 13.

Collisions caused by echo reflection: 4.
Collisions caused by too near obstacles: 9.

Test 2: Collisions: 7.
Collisions caused by echo reflection: 2.
Collisions caused by too near obstacles: 5.

Test 3: Collisions: 10.
Collisions caused by echo reflection: 5.
Collisions caused by too near obstacles: 5.

Test 4: Collisions: 10.
Collisions caused by echo reflection: 4.
Collisions caused by too near obstacles: 6.

There is another robot collision possibility when the robot is going back. As there is no sonar ranging module behind the robot, it may collide.

6. Autonomous Mobile Robot Emmy III

The aim of the Emmy III autonomous mobile robot is to be able to move from an origin point to an end point, both predetermined, in a non-structured environment. We'll do it in steps. First, the robot must be able to move from a point to another in an environment without any obstacle. This environment is divided in cells [6] and a planning system gives the sequence of cells the robot must follow to reach the end cell. This idea was applied in [7], [8]. The second step is an evolving of the first step; the robot must be able to avoid cells that are supposed to have some obstacle in. A sensor system will detect the cells that have to be avoided. This sensor system will use Paraconsistent Annotated Logic to handle information captured by the sensors. The Emmy III structure is:

Sensing system - The robot's environment is composed by a set of cells. On the other hand, the sensing system has to determine the environment with enough precision, but the information captured by the sensors always has an inherent imprecision, which leads to an uncertainty regarding to the position actually the robot is in. In order to manipulate this kind of information, the sensing system is based on the Paraconsistent Annotated Evidential Logic Eτ, which captures the information generated by the sensors using favorable and contrary evidences degrees as seen in the logical controller Paracontrol.

Planning system - The objective is to build a planning system able to determine a path linking an initial point to an end point in a non-structured environment with some obstacles. For this, the environment is divided in cells and the planning system gives the sequence of cells that the robot starting from the initial point reaches successfully the end cell. The first step is to build a planning system for an environment without any obstacle, that is, an environment with all cells free. In the second step the sensing system informs the planning system the cells that have objects in.

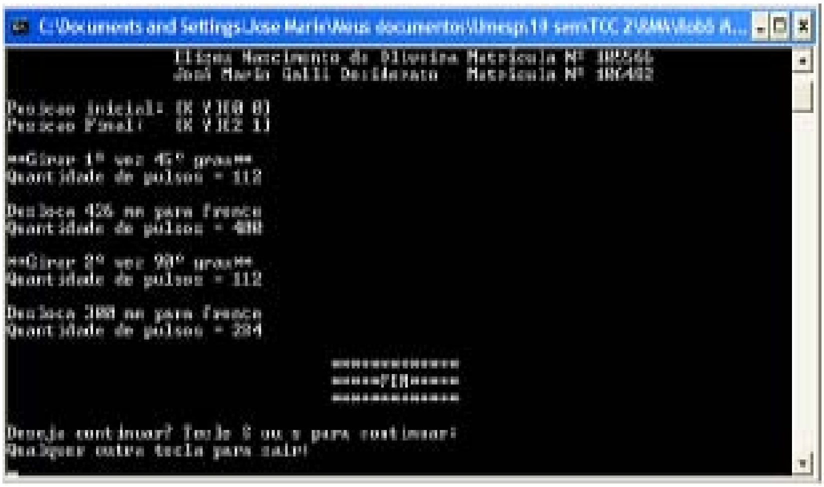
Physical Construction - The Emmy III mechanical part must perform the schedule determined by the planning system. It must know the cell it is in, therefore, a monitoring position makes part of this construction. In the process, for each cell that the robot reaches, the possible error of position should be considered. In the items 7 and 8 is described two Emmy III prototypes where a robot is able to follow a path determined by a planning system in an environment without any obstacle.

7. First Prototype of the Autonomous Mobile Robot Emmy III

The first prototype is composed of a planning system and a mechanical construction. The planning system considers an environment divided in cells. This first version considers all cells free. Then it asks for the initial point and the aimed point.

After that a sequence of movements is given in a screen. Also a sequence of pulses is sent to the step motors that are responsible for moving the physical platform of the robot. So, the robot moves from the initial point to the aimed point.

Figure 22 shows the planning system screen.


C:\Documents and Settings\Jose Maria\Meus documentos\Unesp\[illegible]\TCC 2\[illegible]\Robô A...

```
                    Eliana Nascimento de Oliveira Matricula Nº [illegible]
                    José Maria Galli [illegible]    Matricula Nº [illegible]

Posicao inicial: (X Y)(0 0)
Posicao Final:   (X Y)(2 1)

**Girar 1ª vez 45° graus**
Quantidade de pulsos = 112

Desloca 426 mm para frente
Quantidade de pulsos = 400

**Girar 2ª vez 90° graus**
Quantidade de pulsos = 112

Desloca 300 mm para frente
Quantidade de pulsos = 284

                              *************
                              *****FIM*****
                              *************

Deseja continuar? Tecle S ou s para continuar!
Qualquer outra tecla para sair!
```

Figure 22 – Planning system screen.

The physical construction of the first prototype of the Emmy III robot is basically composed of a circular platform of approximately 286 mm of diameter and two-step motors. The figure 23 shows the Emmy III first prototype. The planning system is recorded in a notebook. And the communication between the notebook and the physical construction is made through the parallel port. A potency driver is responsible to get the pulses from the notebook and send them to the step motors that are responsible for moving the robot.

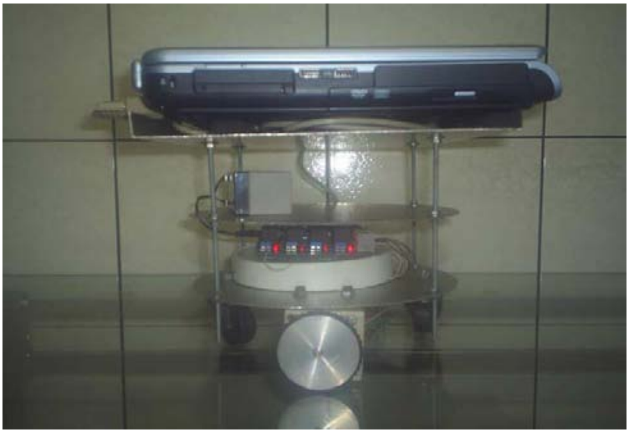
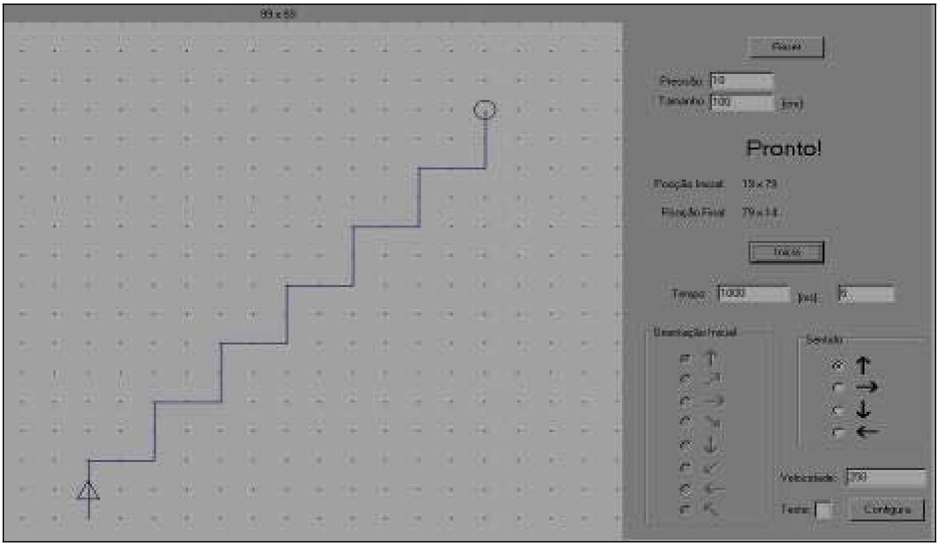


Figure 23. The first prototype of Emmy III robot

8. Second Prototype of the Autonomous Mobile Robot Emmy III

Similarly to the first prototype, the second prototype of the autonomous mobile robot Emmy III is basically composed of a planning system and a mechanical structure. The planning system is recorded in any personal computer and the communication between the personal computer and the mechanical construction is done through a USB port. The planning system considers the environment around the robot divided in cells. So, it is necessary to inform the planning system the cell the robot is in and the aimed cell. The answer of the planning system is a sequence of cells that the robot must follow to go from the origin cell to the aimed cell.

The planning system considers all cells free. Figure 24 shows the screen of the planning system.



89 x 55

Precisão: 10

Tamanho: 100 (cm)

Pronto!

Posição Inicial 13 x 79

Posição Final 79 x 14

Tempo: 1000 (ms): 5

Velocidade: 200

Figure 24. The output of the planning system - Emmy III.

Figure 25 shows the mechanical structure of Emmy III second prototype.

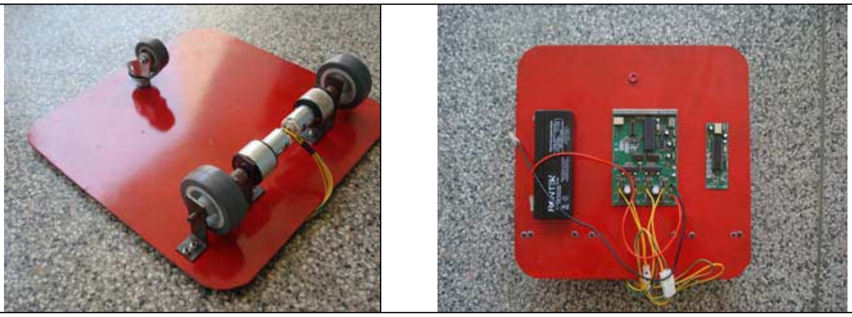


Figure 25. The mechanical structure of the Emmy III second prototype

The planning system considers all cells free. The mechanical construction is basically composed of a steel structure, two DC motors and three wheels. Each motor has a wheel fixed in its axis and there is a free wheel. There is an electronic circuitry on the steel structure. The main device of the electronic circuitry is the microcontroller PIC18F4550 that is responsible for receiving the schedule from the planning system and activates the DC motors. Also there is a potency driver between the microcontroller and the DC motors.

9. Conclusions

In this work we've studied the third prototype of the Emmy III autonomous mobile robot. The main concern is its navigating route planning, i.e. Emmy III is able to move from an origin point to an end point in a non-structured environment.

Two prototypes of this robot' version were built and tested. They are composed of a planning system and a mechanical structure and they were able to move from an origin point to an end point in an environment without any obstacle. Both of them had a satisfactory performance.

The next step is to build a mobile robot with the same characteristics of the described prototypes but adding a sensing system. So we expect that this new prototype will be able to move from an origin point to an end point in a non-structured environment with obstacles. The sensing system also is based on the context of Paraconsistent Annotated Evidential Logic Eτ. We hope to say more in forthcoming papers.

References

[1] ABE, J.M., "Fundamentos da Lógica Anotada" (Foundations of Annotated Logics), in Portuguese, Ph. D. Thesis, University of São Paulo, São Paulo, 1992.

[2] DA SILVA FILHO, J.I., Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos, in Portuguese, Ph. D. Thesis, University of São Paulo, São Paulo, 1999.

[3] ABE, J.M. & J.I. DA SILVA FILHO, Manipulating Conflicts and Uncertainties in Robotics, *Multiple-Valued Logic and Soft Computing*, V.9, ISSN 1542-3980, 147-169, 2003.

[4] TORRES, C. R., Sistema Inteligente Paraconsistente para Controle de Robôs Móveis Autônomos, in Portuguese, MSc Dissertation, Universidade Federal de Itajubá – UNIFEI, Itajubá, 2004.

[5] Datasheet of Polaroid 6500 Series Sonar Ranging Module, 1996.

[6] ELFES, A., Using occupancy grids for mobile robot perception and navigation, Comp. Mag., vol. 22, No. 6, pp. 46-57, June 1989.

[7] DESIDERATO, J. M. G. & DE OLIVEIRA, E. N., Primeiro Protótipo do Robô Móvel Autônomo Emmy III, in Portuguese, Trabalho de Conclusão de Curso, Universidade Metodista de São Paulo, São Bernardo do Campo - SP, 2006.

[8] MARAN, L. H. C., RIBA, P. A., COLLETT, R. G. & DE SOUZA, R. R., Mapeamento de um Ambiente Não-Estruturado para Orientação de um Robô Móvel Autônomo Utilizando Redes Neurais Paraconsistente, in Portuguese, Trabalho de Conclusão de Curso, Universidade Metodista de São Paulo, São Bernardo do Campo - SP, 2006.

[9] JASKOWSKI, S., Um calcul des propositions pour les systems déductifs contradictoires, *Studia Societatis Scientiarum Torunensis*, Sect. A, 1, 57-77, 1948.

[10] DA COSTA, N.C.A., On the theory of inconsistent formal systems, *Notre Dame J. of Formal Logic*, 15, 497-510, 1974.

[11] NELSON, D., Negation and separation of concepts in constructive systems, A. Heyting (ed.), *Constructivity in Mathematics*, North-Holland, Amsterdam, 208-225, 1959.

[12] SUBRAHAMANIAN, V.S. "On the semantics of quantitative Lógic programs "Proc. 4 th. IEEE Symposium on Logic Programming, Computer Society Press,Washington D.C, 1987.

[13] ABE, J.M., Some Aspects of Paraconsistent Systems and Applications, *Logique et Analyse*, 157(1997), 83-96.

[14] DACOSTA, N.C.A., ABE, J.M., DA SILVA FILHO, J.I., MUROLO, A.C. LEITE, C.F.S. Lógica Paraconsistente Aplicada, ISBN 85-224-2218-4, Editôra Atlas, São Paulo, 214 pp., 1999.

[15] DA SILVA FILHO, J.I. & ABE, J. M., Paraconsistent analyzer module, *International Journal of Computing Anticipatory Systems*, vol. 9, ISSN 1373-5411, ISBN 2-9600262-1-7, 346-352, 2001.

[16] ZADEH, L., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" – *IEEE Transaction on Systems*, *Mam and Cybernectics*, vol. SMC-3, No 1, p.p. 28-44, January, 1973.

[17] ABE, J.M. & J.I. DA SILVA FILHO, Simulating Inconsistencies in a Paraconsistent Logic Controller, *International Journal of Computing Anticipatory Systems*, vol. 12, ISSN 13735411, ISBN 2-9600262-1-7, 315-323, 2002.

[18] TORRES, C.R., J.M. ABE & G.L. TORRES, Sistema Inteligente Paraconsistente para Controle de Robôs Móveis Autônomos, Anais do I Workshop Universidade-Empresa em Automação, Energia e Materiais, 5-6 Nov., 2004, Taubaté (SP), Brazil, 2004.

[19] DA SILVA FILHO, J.I. & ABE, J. M., Emmy: a paraconsistent autonomous mobile robot, in Logic, Artificial Intelligence, and Robotics, Proc. 2nd Congress of Logic Applied to Technology – LAPTEC'2001, Edts. J.M. Abe & J.I. Da Silva Filho, Frontiers in Artificial Intelligence and Its Applications, IOS Press, Amsterdan, Ohmsha, Tokyo, Editores, Vol. 71, ISBN 1586032062 (IOS Press), 4 274 90476 8 C3000 (Ohmsha), ISSN 0922-6389, 53-61, 287p., 2001.

[20] DA SILVA FILHO, J.I. & J.M. ABE, Para-Control: An Analyser Circuit Based On Algorithm For Treatment of Inconsistencies, Proc. of the World Multiconference on Systemics, Cybernetics and Informatics, ISAS, SCI 2001, Vol. XVI, Cybernetics and Informatics: Concepts and Applications (Part I), ISBN 9800775560, 199-203, Orlando, Florida, USA, 2001.

[21] ABE, J.M. & J.I. DA SILVA FILHO, Simulating Inconsistencies in a Paraconsistent Logic Controller, International Journal of Computing Anticipatory Systems, vol. 12, ISSN 13735411, ISBN 2-9600262-1-7, 315-323, 2002.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-259

Software Development for Underground and Overhead Distribution System Design

Hernán Prieto SCHMIDT [a], Nelson KAGAN [a], Leandro Rodrigues BARBOSA [a], Henrique KAGAN [a], Carlos Alexandre de Sousa PENIN [b], Alden Uehara ANTUNES [b], Waldmir SYBINE [b], Tânia Paula Ledesma ARANGO [b], Carlos César Barioni de OLIVEIRA [b], Sílvio BALDAN [c] and M. MARTINS [c]
[a] *ENERQ/USP - Av. Professor Luciano Gualberto,Travessa 3, nº 158, Bloco A*
05508-900 - Cidade Universitária - São Paulo, SP - Brazil
[b] *DAIMON - São Paulo, SP - Brazil*
[c] *AES ELETROPAULO - São Paulo, SP - Brazil*

Abstract – This paper presents new methodologies aimed at improving the design of distribution systems from the electrical, the mechanical and the economics viewpoints. The methodologies have been implemented as software solutions fully integrated with AES ELETROPAULO's GIS system.

Keywords. Electricity Distribution, Network Design, Capacity Margin, Optimization.

Introduction

System design, for both expansion and enhancement purposes, was formerly carried out manually on electrical maps drawn on paper. No automatic verifications, electrical or mechanical, could be performed in this case.

At AES ELETROPAULO, most design activity is related to secondary networks, where the individual amount of labor and materials is usually small. However, this activity implies a considerable effort on the part of the Company since the number of projects is high (typically, 1200 per month). Large projects involving the transmission system, primary circuits and distribution substations also occur, but their number is much smaller (around 60 per year).

With no automatic tools for supporting network design, average times for developing complete projects became naturally long. Filtering techniques, such as singling out projects with maximum demand above a pre-specified value, frequently implied developing an unnecessary study or, even worse, overlooking a project that needed a detailed study.

This paper describes the development of a methodology that allows assessing the impact of the connection of new customers on the distribution system, from the electrical, mechanical and economics viewpoints. An important feature of the corresponding computational implementation is its full integration with the GIS platform currently in use at AES ELETROPAULO.

The paper is organized as follows. Sections 2 and 3 describe the most important aspects of the methodology, regarding the electrical, the mechanical and the economics viewpoints, respectively. Section 4 describes the computational implementation and finally Section 5 presents the conclusions of the paper.

1 Electrical Analysis

The most important parameter, which was specifically designed for this application, is the so-called Capacitiy Margin. The capacity margin in a given load point is defined as the maximum load (in kVA) that can be connected to the load point so no technical constraint becomes violated. For distribution networks, the following constraints are enforced:

- maximum loading in each network section (branches);
- minimum voltage at each node;
- maximum loading of the corresponding distribution transformers.

It should be noted that the computed value of capacity margin does take into account the unbalanced state of the network prior to the application of the new load (3- or 4-wire electrical representation) and the number of phases of the new load. For instance, a 2-phase, 3-wire customer can be connected to a 3-phase, 4-wire network in 3 different ways (ABN, BCN or CAN). In this case, 3 values of capacity margin are computed and assigned to the corresponding load point.

The computation of the capacity margin starts with a conventional loadflow analysis. In this case, the current state of the electrical system in the corporate database is of paramount importance. Once the electrical state is known, the capacity margin is computed for all load points in the system, for all possible phase connections of the new load. These values are then stored in specific fields within the GIS database.

At design stage, the technician chooses the load point for a given candidate load and, through a quick query to the GIS database, gets to know the capacity margin at that point. A direct comparison of this value with the specified demand of the new customer allows a decision to be made in no time: connect the new customer without further consideration, or develop a project so a few candidate network modifications can be proposed and analyzed.

The capacity margin at a given load point is automatically updated whenever a change to the network(s) involved in the supply of the load point is made.

Benefits arising from the approach described above are summarized as follows:

- Response times are almost negligible, since no loadflow calculations are performed whenever the capacity margin at a certain load point is queried;
- The two types of wrong decisions described in Section 1 (performing an unnecessary analysis or not performing a mandatory analysis) are automatically eliminated.

In the remaining of this section, the most relevant aspects on the computation of the capacity margin will be discussed.

1.1. Capacity Margin from distribution transformer maximum loading

Figure 1 shows the electrical diagram in the case of a 1-phase new load being connected to a 1-phase distribution transformer. Symbol I_E indicates the current demanded by the existing load (in ampères), while I_N is the corresponding value for the new load being connected, and V_{PN} is the phase-to-neutral voltage on the secondary side of the distribution transformer (volts).

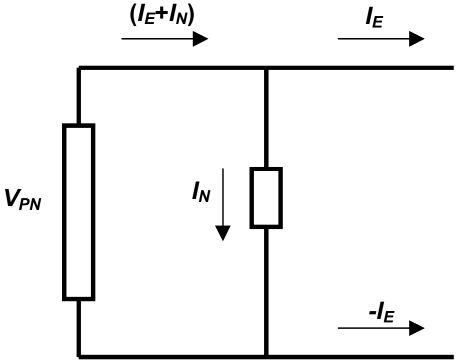


Figure. 1 – Computation of capacity margin

The constraint that guarantees that the transformer's maximum loading will not be exceeded is given by:

$$\left|I_E + I_N\right| = f_O \cdot \frac{S_{rated}}{V_{rated}} \ , \tag{1}$$

where f_O is the allowable overload factor (p.u.), S_{rated} is the transformer's rated power (VA) and V_{rated} is the corresponding phase-to-neutral rated voltage (V). Assuming that the existing load and the new load have both the same power factor (which is not mandatory), the capacity margin *CM* in VA can be derived from Eq. (1):

$$CM = V_{rated} \cdot I_N = f_O \cdot S_{rated} - V_{rated} \cdot I_E \ . \tag{2}$$

In this work, all combinations of distribution transformer connections (1 phase, 2 phases with center tap, open delta, closed delta, and 3 phases) with load connections (1, 2 or 3 phases) have been implemented through the procedure described above.

1.2. Capacity Margin from overhead network maximum loading

In this case, it is assumed that the overhead distribution circuits are all radial. Given the point where the new load is to be connected, identifying all branches connecting the distribution transformer to the load point is a straightforward procedure due to the radial structure of the network. Only these branches will experience a change in current. Among them, the software identifies the so-called *critical branch*, which is the one with the least capacity margin (difference between the branch's rated current and the actual current). The fact that this analysis must be carried out on a per-phase basis introduces an additional difficulty to the problem.

1.3. Capacity Margin for underground networks

In this case, it is assumed that the electrical system and the loads are both balanced. Therefore, a single-phase electrical representation is used. However, the network usually operates in meshed configuration, which requires the use of specific techniques. In this work, the electrical analysis of underground networks is carried out through the nodal formulation (electrical network represented by the nodal admittance matrix).

The starting point in this case is the separate consideration of the existing loads and the new load, which is valid because of the linear characteristic of the nodal formulation. Figure 2 illustrates this approach, in which it is assumed that the new load will be connected to bus j.

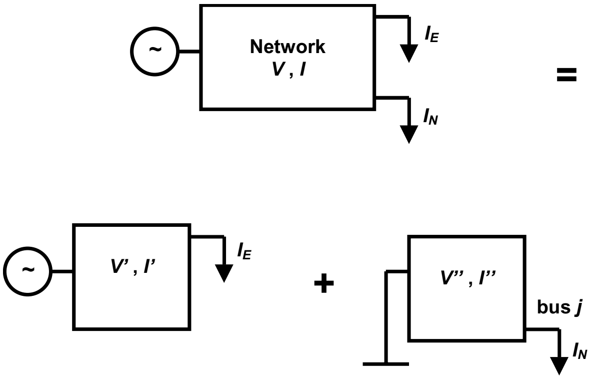


Figure 2 – Separate consideration of the new load at bus j

At generic bus k, the minimum voltage constraint can be written as:

$$V_k = V_k' + V_k'' \geq V_{\min} \ , \tag{3}$$

where V_k' is the voltage at bus k before the connection of the new load, V_k'' is the voltage considering the existence of the new load only (with all voltage generators inactive), V_k is the final voltage (all loads connected), and $V_{\min}$ is the specified minimum voltage.

Voltage V_k'' represents the voltage drop that bus k will experience after the new load is connected, and it can be computed through:

$$V_k'' = -Z_{kj} \cdot I_N \ , \tag{4}$$

where Z_{kj} is the element in row k and column j of the nodal impedance matrix (the inverse of the nodal admittance matrix).

From Eqs. (3) and (4), the maximum load that can be connected at bus j so as to enforce the minimum voltage constraint at any bus k can be computed through:

$$I_N \leq \frac{V_k' - V_{\min}}{Z_{kj}} \ . \tag{5}$$

An identical approach can be used to determine the impact of the maximum branch loading constraint. In this case, the generic branch connecting busses *r* and *s* is considered:

$$I_{rs} = I'_{rs} + I''_{rs} \leq I_{\max} \ , \tag{6}$$

and the current caused solely by the new load at bus *j* (I''_{rs}) is given by:

$$I''_{rs} = \frac{V''_r - V''_s}{z_{branch\_rs}} = -I_N \frac{Z_{rj} - Z_{sj}}{z_{branch\_rs}} = I_N \frac{Z_{sj} - Z_{rj}}{z_{branch\_rs}} \ , \tag{7}$$

where z_{branch_rs} is the impedance of branch *rs*. From (6) and (7) the maximum value for I_N corresponding to the maximum branch loading constraint can be computed:

$$I_N \leq \frac{I_{\max} - I'_{rs}}{Z_{sj} - Z_{rj}} \cdot z_{branch\_rs} \ . \tag{8}$$

2 Mechanical Analysis

The software tool allows the computation of the total bending moment in any pole. In the computation of the bending moment, the mechanical stress from the following components is taken into account:

- power conductors;
- conductors from third-party users such as telephone utility and cable TV;
- other equipment directly attached to the pole, such as distribution transformers.

If the total bending moment is greater than the pole's maximum allowable bending moment, the designer can choose one of the following solutions:

- replace the pole with another pole with sufficient mechanical capability;
- install an anchor pole with guys (in this case, the total bending moment is computed again considering the existence of the anchor pole).

In the case of underground networks, the software tool allows the computation of the pulling stress required to install the insulated cables, as well as the pressure applied by the cables to the inner surfaces of curves within the ducts.

3 Evaluation and Selection of Alternatives

When a new load is added to an existing distribution network, it is possible for the network not to be able to withstand the extra power requirement. Depending on the type of the electrical constraint that is being violated, the designer can take the following corrective actions:

- Distribution transformer maximum loading exceeded: (1) distribution transformer replacement, and (2) circuit splitting with additional distribution transformer;
- Branch maximum loading and/or maximum voltage drop exceeded: (1) cable replacement, and (2) circuit splitting with additional distribution transformer.

In all cases, the designer has to make decisions as to what solution to implement. In this work, two different conceptual approaches to this end were implemented. Both will be described below.

The first approach uses both capital (investment) and operational costs associated with every alternative. Operational costs mean in this case values assigned to technical losses, voltage profile and non-distributed energy. This approach is more suited to situations where budget constraints are normally enforced: the decision to be made should be based primarily on cost minimization.

In the second approach, benefits arising from the modifications applied to the network are also taken into account. As in the preceding case, benefit values are assigned to technical losses, voltage profile and non-distributed energy. Cost-benefit analysis such as the present one is more suited to situations where capital return is pursued.

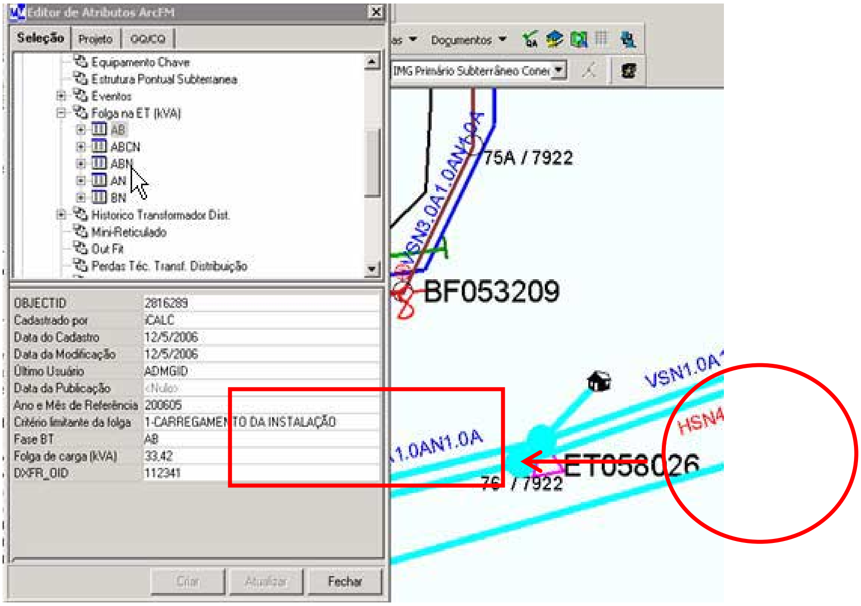
In both cases, the electrical network must be evaluated from the technical point of view prior to the modifications indicated by the designer.

4 Computational Implementation

4.1 Capacity Margin

As defined in Section 2, the capacity margin in a given load point is the maximum load (in kVA) that can be connected to the load point so no technical constraint becomes violated. Due to the specific characteristics of overhead and underground distribution networks, the computation of the capacity margin was implemented as two different computational modules.

Figure 3 shows an example of capacity margin query in an overhead secondary distribution circuit.



OBJECTID	2816289
Cadastrado por	iCALC
Data do Cadastro	12/5/2006
Data da Modificação	12/5/2006
Último Usuário	ADMGID
Data da Publicação	<Nulo>
Ano e Mês de Referência	200605
Critério limitante da folga	1-CARREGAMENTO DA INSTALAÇÃO
Fase BT	AB
Folga de carga (kVA)	33,42
DXFR_OID	112341

Figure 3 – Capacity Margin query

4.2 *Mechanical Analysis*

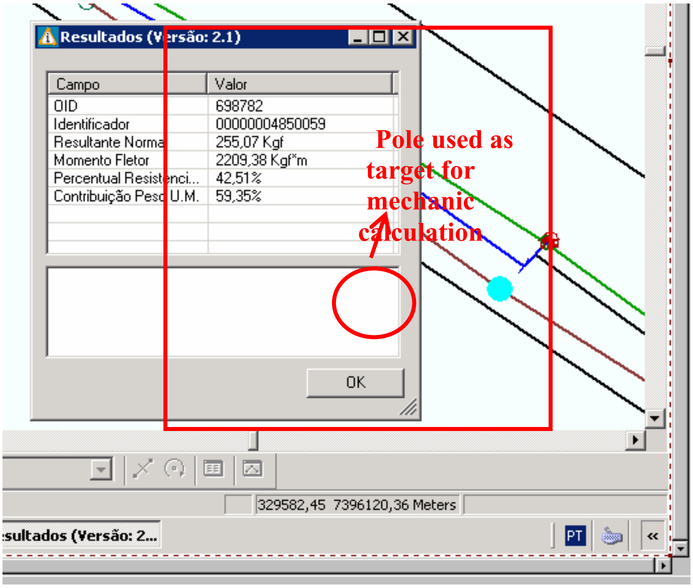
The calculations implemented in connection with mechanical analysis are summarized as follows:

- Force arising from conductors existing in the primary and/or secondary network;
- Moments resulting from the action of conductors and other equipment such as distribution transformers;
- Evaluation of anchor poles and guys in case that any mechanical parameter exceeds its corresponding maximum value.

Figure 4 shows an example of a mechanical calculation.

4.3 *Selection of Best Design Alternative*

This module was developed and implemented bearing in mind a full integration with the GIS platform currently in use at AES ELETROPAULO. After the designer entered all network modifications, the module executes a technical analysis, whereby all relevant parameters are estimated: losses, voltage profile and loading. The economics analysis is carried out thereafter, yielding the total cost associated with the alternative under consideration and its basic components (loss, voltage profile, etc.). Results can be analyzed through specific reports that enable the designer to complete the decision-making process.



Resultados (Versão: 2.1)

Campo	Valor
OID	698782
Identificador	00000004850059
Resultante Norma	255,07 Kgf
Momento Fletor	2209,38 Kgf*m
Percentual Resistenci...	42,51%
Contribuição Peso U.M.	59,35%

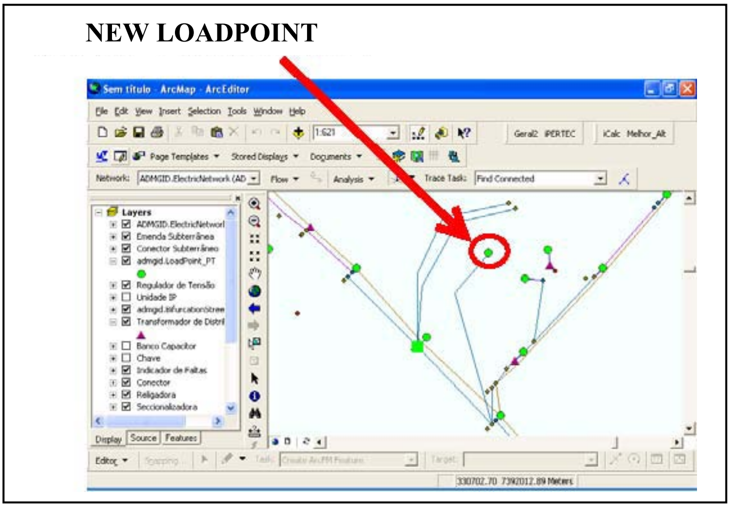
OK

329582,45 7396120,36 Meters

Pole used as target for mechanic calculation

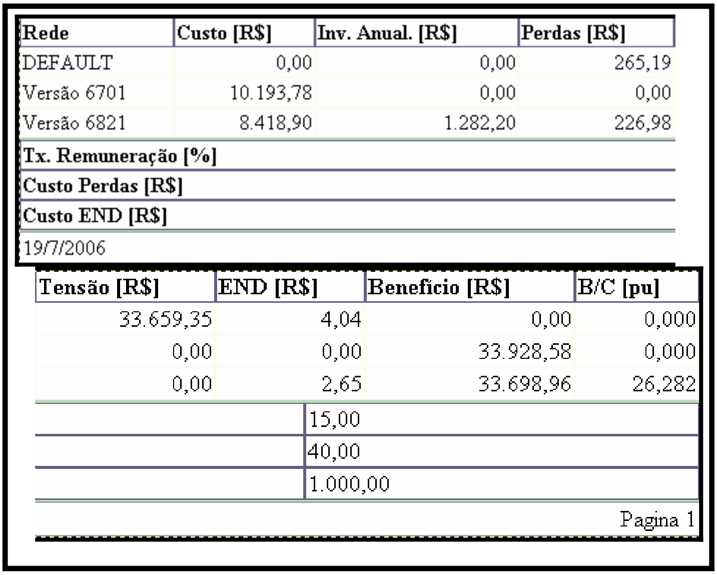
Figure 4 – Example of mechanical calculation

Figure 5 shows the construction of a network extension aimed at connecting a new load point, while Figure 6 shows a condensed report on the analysis of the best alternative, including costs, benefit and cost-benefit ratio. It should be noted that the total benefit is estimated by comparing the network's new operational condition with the former condition (before the introduction of design modifications).



NEW LOADPOINT

Figure 5 – Construction of a network extension



Rede	Custo [R$]	Inv. Anual. [R$]	Perdas [R$]	Tensão [R$]	END [R$]	Beneficio [R$]	B/C [pu]
DEFAULT	0,00	0,00	265,19	33.659,35	4,04	0,00	0,000
Versão 6701	10.193,78	0,00	0,00	0,00	0,00	33.928,58	0,000
Versão 6821	8.418,90	1.282,20	226,98	0,00	2,65	33.698,96	26,282
Tx. Remuneração [%]	15,00						
Custo Perdas [R$]	40,00						
Custo END [R$]	1.000,00						
19/7/2006							Pagina 1

Figure 6 – Selection of the best alternative

4.4 Integration with GIS

Another feature of the software tool lies in the direct integration with *Interplan* [2, 3], a software solution aimed at developing mid-term planning studies. Figure 7 shows a primary distribution network that will receive a major network extension. This extension is designed using *Interplan*'s tools and the final result is illustrated in Figure 8.

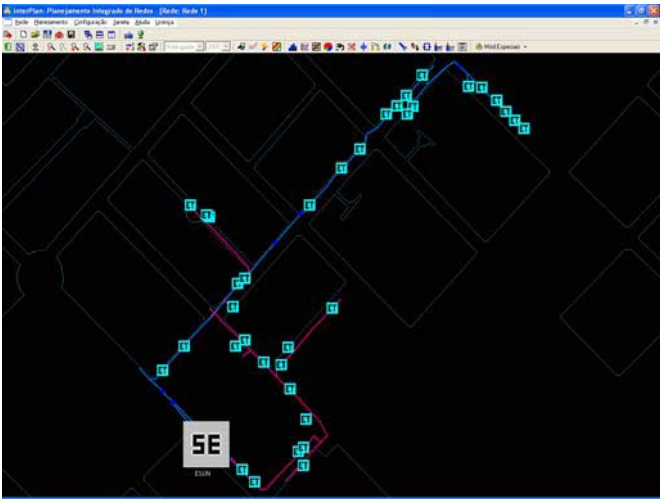


Figure 7 – Existing primary network

The network extension can be imported directly to the design environment as illustrated in Figure 9.

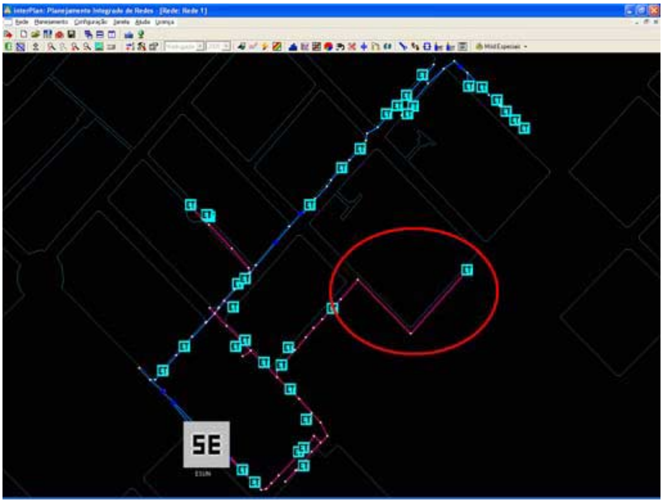


Figure 8 – Network extension designed with Interplan

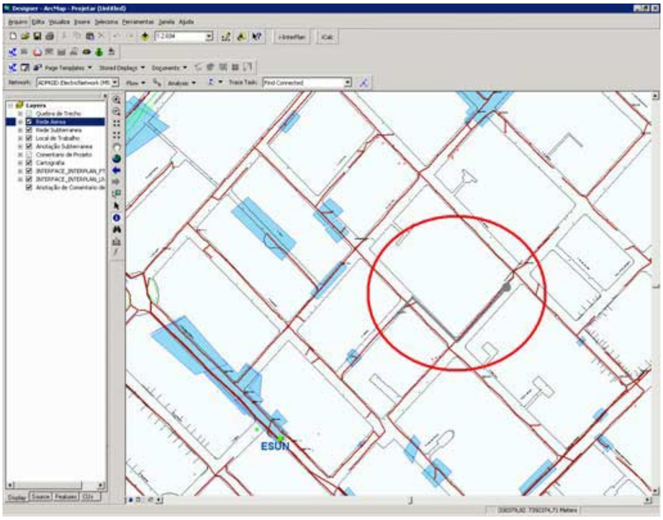


Figure 9 – Network extension imported in the design environment

5 Conclusions

This paper has presented a methodology and the corresponding computational implementation for aiding the design of distribution networks. An important feature of the methodology lies in the new concept of *Capacity Margin*, by which the maximum load that can be connected at any existing point in the network is computed offline and made available easily to network designers. Besides optimizing the design cycle, this concept has also eliminated the possibility of making wrong decisions, which could arise through the former approach (based on filtering designs relying solely on the maximum demand informed by customers).

Another important feature of the software tool is its straightforward integration with the GIS platform currently in use at AES ELETROPAULO. In this case, the software tool also benefits from the resources available from other external applications that are integrated to the GIS platform as well.

References

[1] W. D. Stevenson Jr.: Elementos de Análise de Sistemas de Potência, 2a Edição em Português, McGraw-Hill, 1986.

[2] H. P. Schmidt, A. U. Antunes: Validação dos Programas Computacionais de Cálculo Elétrico (iCalc e iRet) e Cálculo de Folga. Relatório Enerq/USP, São Paulo, 04/2005.

[3] H. P. Schmidt, A. U. Antunes: Especificação do módulo para cálculo de fluxo de potência subterrâneo - programa iRet. Relatório Enerq/USP, Ano 1 - Etapa 6, Projeto GIS Subterrâneo.

[4] OLIVEIRA, C. C. B. de; SCHMIDT, H. P.; KAGAN, N.; ROBBA, E. J. Introdução a sistemas elétricos de potência - componentes simétricas. 2. ed. São Paulo, Brasil: Edgard Blücher, 1996. v. 1. 467 p.

[5] Méffe, A.; Kagan, N.; Oliveira, C. C. B; Jonathan, S.; Caparroz, S. L.; Cavaretti, J. L., "Cálculo das Perdas Técnicas de Energia e Demanda por Segmento do Sistema de Distribuição". XIV Seminário Nacional de Distribuição de Energia Elétrica, Foz do Iguaçu, Brasil, Novembro de 2000.

[6] ORSINI, L. Q. Curso de circuitos elétricos. São Paulo, Edgard Blücher, 1993-4. 2v.

[7] JARDINI, J. A. e outros. Curvas Diárias de Carga – Base de Dados Estabelecida com Medições em Campo, CIRED, 1996.

[8] JARDINI, J. A. et all. Residential and Commercial Daily Load Curve Representation by Statistical Function for Engineering Studies Purposes, CIRED, 1995.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-936-3-270

Distribution Transformer Technical Losses Estimation with Aggregated Load Curve Analytical Methodology and Artificial Neural Network Approach Implementation

Adriano Galindo LEAL [a], Jose Antonio JARDINI [b] and Se Un AHN [c]
[a] *Elucid Solutions - Av. Angélica, 2318, 5º andar - 01228-904 - São Paulo,SP -Brazil*
[b] *University of São Paulo - EPUSP - Av. Professor Luciano Gualberto,Travessa 3, nº 158 - 05508-900 - Cidade Universitária - São Paulo, SP - Brazil*
[c] *CPFL Energy - Rodovia Campinas Mogi-Mirim km 2,5 -13088-900 – Campinas, SP-Brazil*

Abstract. The paper proposes an analytical methodology and three different Artificial Neural Network alternatives using a multi layer perceptron (MLP) in order to estimate technical losses in distribution systems. For each transformer a load curve is set by aggregated customer load curves, and then losses are estimated by transformer's daily load curve stratified, with the proposed methodologies. Theoretically, it can reach a different load curve for each transformer then consequently distinguished losses either. The losses are estimated for distribution transformer, but the methodology can be used on each segment involved in the distribution system (secondary and primary network, distribution transformers, HV/MV transformers). This is done by using the network's data, the consumer's monthly energy consumption data and the typical load curves by class of consumer and type of enterprise developed.

Keywords. Distribution Transformer Losses, Neural Networks, Information Systems, Geographic information systems, Gaussian distributions.

Introduction

Estimating technical losses of a distribution system is considered one of most important activities of every electrical utility engineering department. This is due to precise loss estimation, the utility can manage its commercial losses, and it mirrors the quality of distribution system, consequently the adequate supply of necessary investments for future years.

The methodology was developed aiming distribution transformers, but it can be used to other part of distribution system. The transformer is the most important and complex gear of distribution system and the most expensive too.

In [1], [2], [3] and [4], detailed academic studies characterizing the consumers daily load profiles were reported. Demand measurements of several types of consumers (residential, commercial and low voltage industrial consumers), were carried out. The consumers' daily load curves were set to record 96 points (i.e. average active power was recorded at intervals of 15 minutes).

For each consumer and distribution transformer, about 15 daily profiles were considered and the mean (*m*) and standard deviation (*s*) profiles were determined and set to characterize the consumer and the distribution transformers.

For ease of manipulation, the demand values were normalized (per unit) by the monthly average demand (energy divided by 30*24 hours). Figure 1 shows the *m, s* profiles of a distribution transformer.

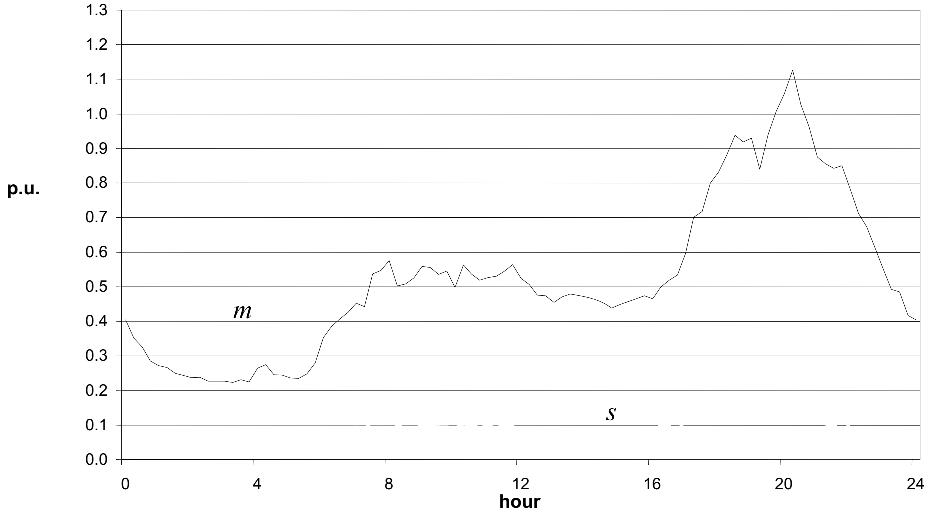


Figure 1. Mean and Standard deviation daily curve of a transformer

A procedure to aggregate (sum) the consumers' demand in a distribution transformer was also developed. Consider a distribution transformer with *r* consumers of the *p* type and *n* of the *q* type. The aggregated demand values (in kW) were calculated using Eq. (1):

$$m_a(t) = \sum_{i=1}^{r} m_{pi}(t) + \sum_{j=1}^{n} m_{qj}(t)$$
$$s_a{}^2(t) = \sum_{i=1}^{r} s_{pi}^2(t) + \sum_{j=1}^{n} s_{qj}^2(t) \qquad (1)$$

The demand value within an interval t is assumed to follow a Gaussian distribution, so the figure with a certain non-exceeding probability can be calculated by:

$$p_g(t) = m(t) + k_g * s(t) \qquad (2)$$

For example: when k_g=1.3, the probability $p_{1.3}$ so as not to be exceeded is 90%.

Figure 2 shows a set of 11 typical profiles with probabilities of 2.5%, 10%, 20%… 90% and 97.5%. These sets of profiles were used in [4] to evaluate the distribution transformer's loss of life due to loading. The same set of profiles is used here to evaluate the losses in distribution transformers.

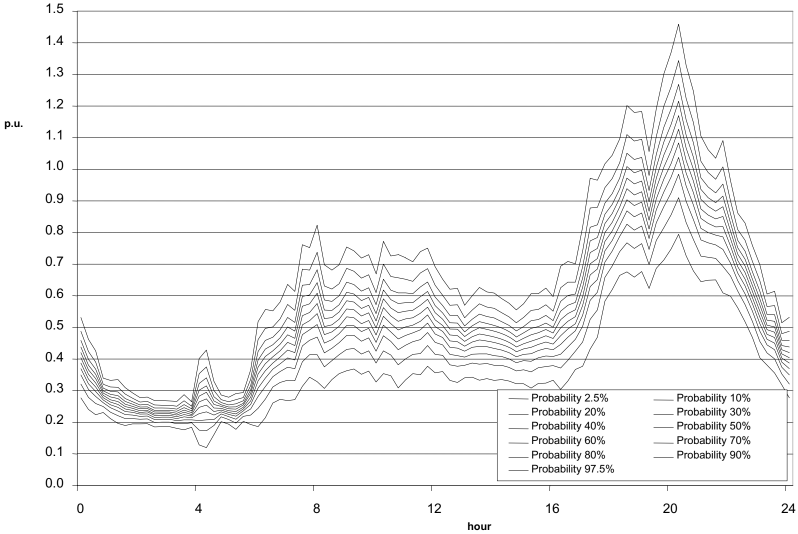


Figure 2. Transformer's daily load curve stratified in 11 curves

The expected losses in the other parts of the distribution system (primary and secondary networks, HV/MV transformers) can be evaluated through a similar approach to thereafter find the global losses of the interconnected system. In this article, the expected cost of losses wasn't a particular matter of analysis, because it includes previous long term investment planning, politics and energy costs, different for each company. But it is a recommendable subject for future works.

1. Distribution Transformer Losses Calculation – General Approach

1.1. Equations

Losses in a distribution transformer are actively produced when the current flows through the coils. It is also due to the magnetic field alternating in the core. They can be divided into no-load losses and on-load losses [8], [9].

No-load losses (L_{nl}) are caused by the magnetizing current needed to energize the core of the transformer and are mostly composed of hysteresis and eddy current losses. It does not vary according to the transformer loading; instead it varies with the voltage and may be considered constant for calculation purposes, taking into careful consideration all inaccuracies in the losses calculation procedure.

On-Load Losses (L_{wl}) vary according to the transformer loading. It comprises heat losses and winding eddy current losses in the primary and secondary conductors of the transformer and other stray losses. Heat losses, also known as series losses or winding I^2R losses, in the winding materials contribute with the largest part of the load losses. The computations in this work will only consider such losses.

As $p_g(t)$ represents the transformer loading in the interval *t* of one profile, then, the series losses L_g (t) can be written as:

$$L_g(t) = r_{pu} * S_r * \left(\frac{p_g(t)}{S_r} \right)^2 = k * \left(\frac{p_g(t)}{S_r} \right)^2 \qquad (3)$$

Where: *k* represents the series losses of the rated power (S_r).

Eq. (3) is applicable to all the 11 profiles g. For instance, if *g*=20%, then this profile can be the selected representative of all the profiles with probability 15 to 25%, meaning a participation factor (*pf*) of 10% (*pf*=0.1).

Note in Table 1 that nine profiles have *pf*=0.1, whereas two profiles have *pf*=0.05.

Thus, the total average series losses (L_{wl}) can be expressed as:

$$L_{wl} = k * \sum_{g=1}^{11} pf_g * \frac{1}{96} * \sum_{t=1}^{96} \left(\frac{p_g(t)}{S_r} \right)^2 \qquad (4)$$

The total distribution transformer losses being:

$$L_T = L_{nl} + L_{wl} \qquad (5)$$

Table 1. Stratification Profiles

Representative Range (%)	Probabilities pf (%)	kg
0 – 5	5	-1.96
5 – 15	10	-1.28
15 – 25	10	-0.84
25 – 35	10	-0.525
35 – 45	10	-0.255
45 – 55	10	0
55 – 65	10	0.255
65 – 75	10	0.525
75 – 85	10	0.84
85 – 95	10	1.28
95 – 100	5	1.96

1.2. Data Handling

From the Distribution Utility database, the following information is used: *m* and *s* characteristic profile (in p.u. of the monthly average demand) of all consumers' type; the transformer parameters (rated power, series and no-load rated losses) and the amount, type and energy consumption per month (E) of the consumers connected to the transformers.

The *m* and *s* profiles (in kW) of each consumer (in a transformer) are calculated multiplying the *m* and *s* values (in p.u.) representative of the type of consumer by its average demand (E/720).

The *m* and *s* profiles of the transformer are obtained through Eq. (1), hence the 11 profiles depicted in Figure 2.

Also, by using Eq. (4) the load losses can be evaluated. Such losses are then added to the no-load losses. Figure 3, discloses the procedure employed named here as Analytical Procedure. This efficient procedure was carefully applied to the 61485 set of transformers of a utility in Brazil and the results will be detailed here.

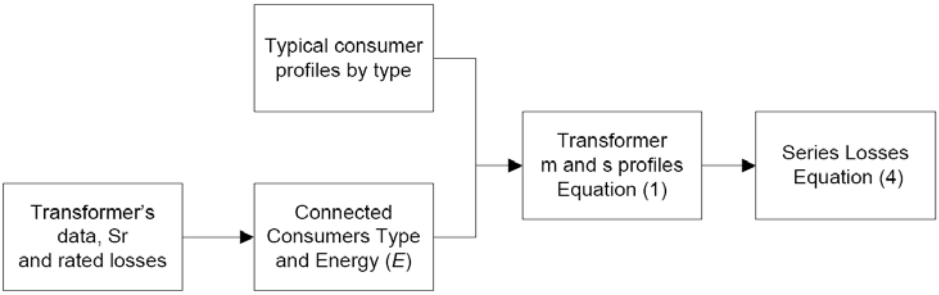

Typical consumer profiles by type
Transformer's data, Sr and rated losses
Connected Consumers Type and Energy (*E*)
Transformer m and s profiles Equation (1)
Series Losses Equation (4)

Figure 3. Load Losses Calculation

1.3. Application

Seven daily load profiles of 57 distribution transformers were recorded at CPFL. The average series losses for every load profile as well as the seven-day average losses were determined.

The *m, s* curves of each transformer were also determined from the seven-day measured load profiles curves. The series losses for the 57 transformer were otherwise determined using Eq. (4).

The distribution of the errors between those two calculations is showing on Figure 4. As it can be seeing, the errors are small with a mean of 0.3%, which means that the Analytical Procedure led to right results.

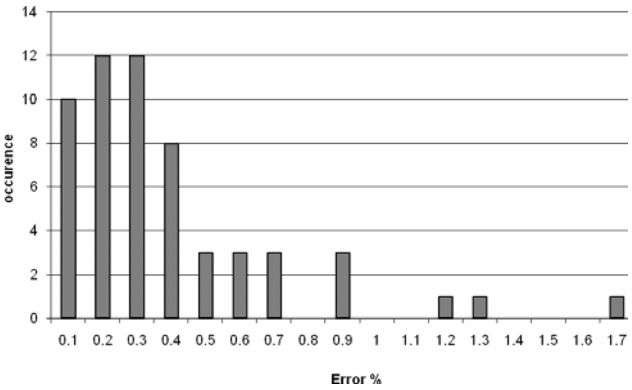


Figure 4. Error Distribution.

Thus the distribution transformer losses can be calculated using the existing information on the utilities database. The calculations were performed for a set of 61485 transformers of a utility applying the Analytical Procedure and the losses assigned as “true values”. On this framework “true values” means that these calculated values are considerate the most close to the real measured value.

2. Distribution Transformer Series Losses Calculation – Basic ANN Approach

2.1. ANN Model

The necessary inputs for the ANN (Artificial Neural Network) model used are the *m* and *s* profiles of the transformers. The output being the series losses divided by *k/96* (4).

The quantity of neurons and layers are defined by trial and error tests in the MLP (multilayer perception) model. The supervisioned and back propagation training is also chosen.

The available data and calculated values of the losses according to section II are partially used for the training stage and partially used for testing the training efficiency.

2.2. Clustering

The set of 61485 transformers daily profiles (m, s), obtained according to aggregation Eq. (1), were normalized (per unit) by the transformer average demand and formally submitted to a cluster analysis.

Next, it was specified to group the transformers into 10 clusters using as criterion the Euclidian distance.

This means that transformers with similar profiles are assigned to the same “cluster box”. Figure 5 and 6, show the average (m, s) curves of two different clusters.

It can be seen that the mean profiles differ because one pertains to typical commercial/industrial loads (Figure 5) whereas the other shows a peak characteristic (at 20hs) of a residential load (Figure 6).

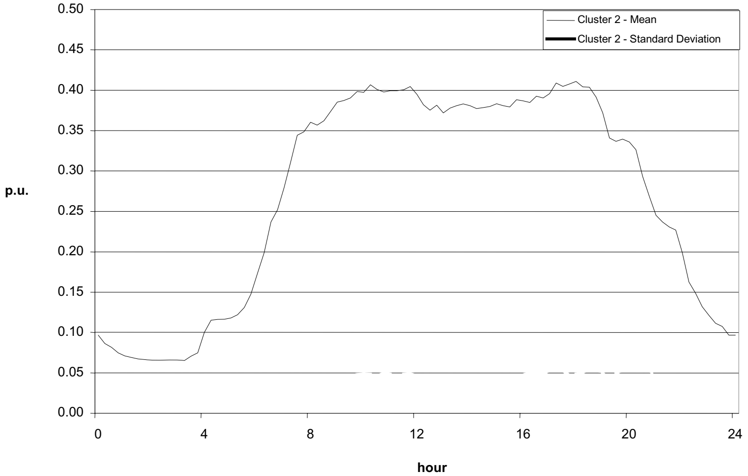


Figure 5. Result of Cluster 2 – Frequency: 3204 (Commercial and Industrial Area).

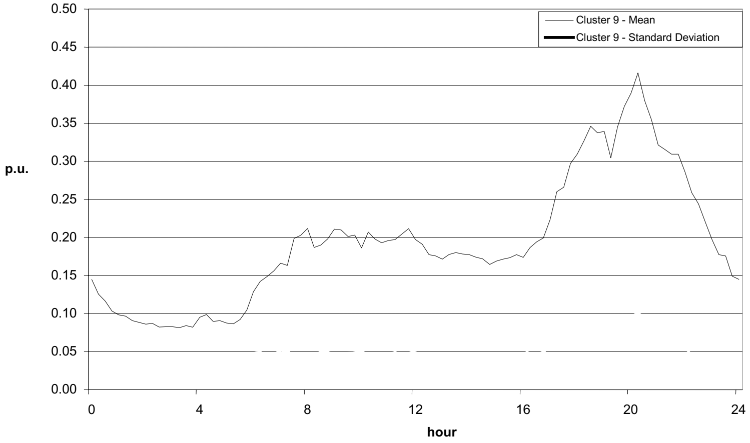


Figure 6. Result of Cluster 9 – Frequency: 14698 (Residential Area).

Table 2, shows the characteristic parameters of each cluster. Clusters 1 through 6 looks like Figure 5, whereas Clusters 7 through 10 like Figure 6.The profile patterns in the 10 clusters correspond to the two load types previously mentioned. The obvious difference in clusters is due to the peak value.

Table 2 Characteristics of the clusters

Cluster	Shape	Frequency	Peak Value p.u.
1	Commercial / Industrial	26120	0.07
2	Commercial / Industrial	3204	0.38
3	Commercial / Industrial	197	0.97
4	Commercial / Industrial	1509	0.67
5	Commercial / Industrial	109	2.77
6	Commercial / Industrial	1931	0.28
7	Residential	9820	0.74
8	Residential	596	1.77
9	Residential	14698	0.42
10	Residential	3301	1.13

Since the profiles are normalized by the transformer's rated power, apart from the shape, the main characteristic of the cluster will be the loading state. The purpose of the clustering procedure was to evaluate if better results are achieved by training only one ANN for each cluster or one for all the transformers.

From the *clusterization* process it can be concluded that:

- 42.5% of the transformers, represented by cluster 1, are performing under extremely low loading levels. This cluster represents transformers having commercial/residential loads and which could be used for future reallocations during the system's expansion program;

- 46.2% of the transformers have typical load conditions of residential type areas (Clusters 7 through 10);
- 1.1% of the transformers, those pertaining to Clusters 5 and 8, are more loaded;

The remaining 10.2% represent transformers of commercial areas.

2.3. Results

The training vector and the test vector are formed by group of inputs and one output, constituted by:

- 24 points of the transformer m curve (in p.u., 1 point per hour);
- 24 points of the transformer s curve (in p.u., 1 point per hour);
- The value of $L_{wl} * \frac{96}{k}$, calculated through the method described in Section II, constitutes the output variable.

The parameters used in all the simulations of the ANN model, are presented in Tables 3 and 4. Table 5 shows the amount of elements in the training and test vectors used.

2.4. Architecture

Table 3: Artificial Neural Network Topology and Parameters

Layers	4
Input Layer Neurons	48
Second Layer Neurons	35
Third Layer Neurons	24
Output Layer Neurons	1

2.5. Training

Table 4 ANN method parameters.

Internal Interactions	**Total Interaction**	**Tolerance (%)**
9	9000	0.15

Table 5 Distribution of Transformers within the test and training Vectors

Cluster	**Total**	**Training**	**Test**
1	26120	2124	1590
2	3204	1283	1601
3	197	80	98
4	1509	605	754
5	109	45	54
6	1 931	773	965
7	9820	1101	1099
8	596	239	297
9	14 698	1101	1099
10	3 301	1321	1650
E000*	11432	3009	1990

Note: One ANN, denominated E000, was trained using a subset of each cluster following its proportion.

It was evaluated the errors in each cluster as well as the percent of cases with errors below 10%, here called "error index". The error indexes in all clusters are shown in Table 6. It can be observed that the best result for each cluster was readily obtained using the specifically trained ANN and sometimes the ANN E000. The error in about 92.5% of the transformers is below 10%.

Table 6 Percentage of Transformers with Error Less than 10%

Cluster	Results using the E000 Network	Results using the Neural Network specifically trained for each Cluster
1	7.0	86.9
2	97.6	95.2
3	85.8	70.1
4	96.7	86.6
5	56.9	67.0
6	53.0	92.5
7	99.5	99.2
8	91.6	86.2
9	78.0	97.8
10	99.1	94.6
All Transformers	54.4	92.5

Although the results can be considered adequate, the procedure involves too many operations due to the number of layers and neurons. So, the overall computation is more time consuming than the analytical procedure. The next section presents some ANN alternative architectures through which it can be achieved similar results. Although the accuracy is reduced they need less input data.

3. Less Time Consuming ANN Approach

Two other architectures are put forward in this paper to enhance the calculation time.

3.1. Alternative 1

The training vector and the test vector are formed by group of inputs and one output, constituted by:

- 4 inputs of the transformer *m* profile at 3, 14, 19 and 21 hours;
- 2 inputs of the transformer *s* profile at 12 and 18 hours.
- The value of $L_{wl} * \frac{96}{k}$, constitutes the output variable.

The parameters used in all the simulations of the ANN model, are the same as those shown in Tables 3 and 4, except that the layer neurons are 6 for the first layer, 18 and 10 for the hidden layers, and 1 for the output layer. The same clusterization result on the initial approach was used. For comparison matters, the same amount of elements in the training and test vectors were used (see Table 5). The error indexes (percent of cases with errors below 15%) in all clusters are shown in Table 7.

Table 7 Percentage of Transformers with Error Less than 15%

Cluster	Results using the E000 Network	Results using the Neural Network Specifically trained for each Cluster
1	31.7	64.4
2	94.3	19.36
3	86.3	0.51
4	95.9	9.0
5	56.0	12.8
6	80.6	16.2
7	99.7	25.4
8	95.0	6.6
9	96.5	12.6
10	99.2	32.1
All Transformers	69.8	37.3

The outcomes using the E000 network are not good (69.8%), however when applying the best trained Neural Networks (the specifically trained for the cluster or E000) to each cluster, 83.2% from the total estimations were obtained with errors below 15%. The results still poor, suggesting that probably a new clusterization should have been done.

3.2. Alternative 2

In this practical approach, all consumers were classified into four types (Figure 7 to 10). With this classification, it was possible to calculate the amount of each type of consumer connected to the transformer and its total energy consumption.

A new clustering process was performed, and the results are presented in table 8, where Qty is the mean amount of the consumer type in the cluster and kWh is the mean consumption of the consumer type in the cluster.

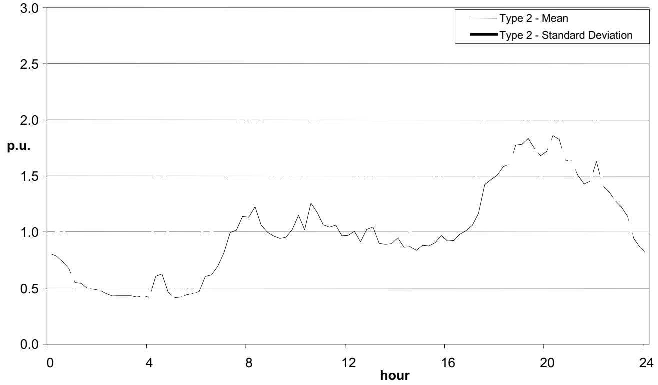


Figure 7. Consumers' daily load curves Type 1 (Residential)

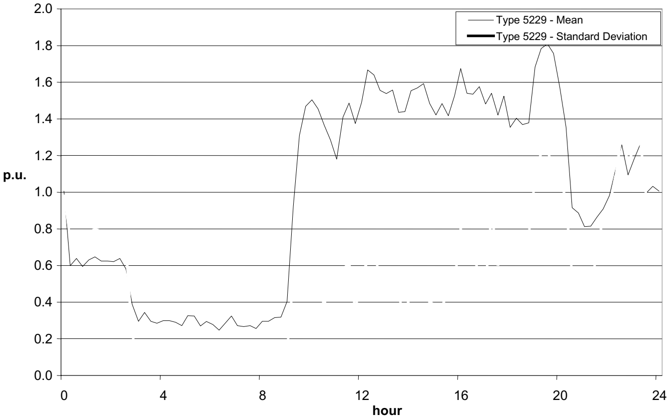


Figure 8. Consumers' daily load curves Type 2 (Industrial)

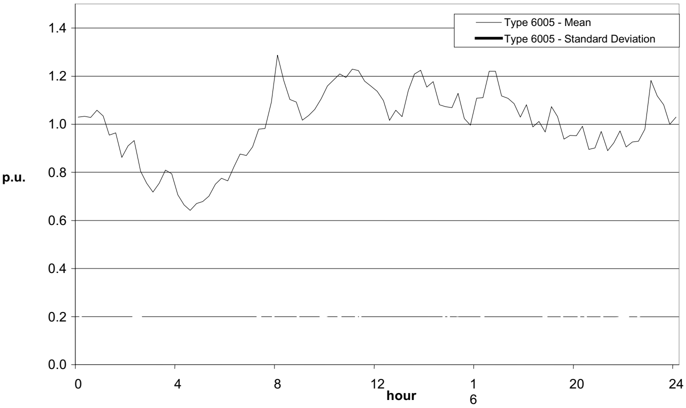


Figure 9. Consumers' daily load curves Type 3 (Flat)

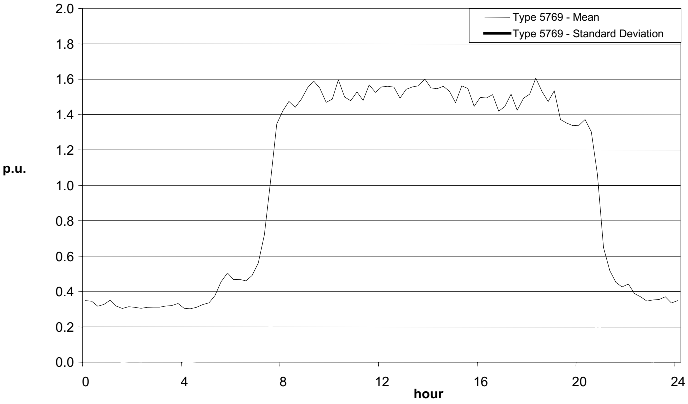


Figure 10. Consumers' daily load curves Type 4 (Commercial)

Table 8 Characteristics of the clusters

Cluster		Type 1		Type 2		Type 3		Type 4	
ID	**Freq.**	**Qty**	**kWh**	**Qty**	**kWh**	**Qty**	**kWh**	**Qty**	**kWh**
1	30898	6	974	1	189	1	68	1	392
2	12162	47	8592	1	460	1	108	2	807
3	8026	87	16617	2	803	1	188	3	1376
4	3237	136	27792	3	1401	1	354	4	2251
5	3194	10	1948	2	774	1	207	6	6847
6	1542	11	2254	4	7286	1	243	3	1462
7	1197	27	5632	6	3796	1	779	18	22414
8	610	200	49216	2	1672	1	342	4	2905
9	347	29	6225	8	22352	1	540	9	5220
10	272	21	4653	3	1541	2	16016	5	3160

In this approach, the training vector and the test vector are formed by group of inputs and one output, constituted by:

- 8 inputs representing the aggregate quantity of consumers and consumption of each consumers type;
- 1 input representing the transformer nominal power;
- The value of $L_{wl} * \frac{96}{k}$, constitutes the output variable.

Again, the parameters used in all the simulations of the ANN model are the same shown in tables 3 and 4, except that the layer neurons are 9 for the first layer, 16 and 8 for the hidden layers, and 1 for the output layer.

Table 9 shows the test and training vectors size used. The error indexes (percent of cases with errors below 15%) in all clusters are presented in Table 10.

Table 9 Distribution of Transformers within the test and training Vectors by Cluster.

Cluster	Total	Training	Test
1	30898	3507	1993
2	12162	3501	2499
3	8026	3001	2999
4	3237	1296	1618
5	3194	1279	1596
6	1542	618	770
7	1197	480	598
8	610	245	304
9	347	140	173
10	272	110	135
E000	61485	7496	2959

Table 10 Percentage of Transformers with Error Less than 15%

Cluster	Results of the E000 Network	Results of the Neural Network application specifically trained for each Cluster
1	17.66	35.3
2	64.4	76.8
3	87.3	80.0
4	85.4	80.4
5	54.2	46.4
6	49.0	36.1
7	52.2	57.2
8	69.2	78.2
9	47.0	8.4
10	52.6	61.0
E000	44.4	53.4

When applying the best trained Neural Networks to each cluster, 55.8% from the total estimations were obtained with errors below 15%.

4. Extension to Other Parts of the Distribution System

The same procedure used for the distribution transformers may be applied to evaluate the series losses in the secondary and primary network as well as for the HV/MV transformers. For instance, for a section of a primary feeder the *m* and *s* curves of transformers beyond this section can be aggregated using Eq. (1). A specific ANN and test procedure shall be carried out to train and calculate the primary feeder losses.

5. Analysis of the Results (ANN approach)

Using the described ANN architectures in items 3 and 4, it can be determined the global loss (all the distribution transformers), the values are showed in table 11.

Table 11 Comparison of Results

	Global Loss (kWh)	Error (%)
Analytical Method	650745	Reference
Initial ANN Architecture (48 input layer neurons)	714165	9.7%
Alternative 1 ANN Architecture (9 input layer neurons)	482368	-25.9%
Alternative 2 ANN Architecture (6 input layer neurons)	502158	-22.8%

6. Conclusion

It can be concluded that: as the load curve was better represented the initial ANN architecture should be the best one obtained. The errors were less than 10% in 92.5% of the transformers. The disadvantage of this method is that the amount of mathematical operations necessary to arrive at this result is greater than that needed in the analytical method.

The second and third architectures have had global errors of 22.8 and 25.96%, respectively. It was less accurate, however its processing time was less as that architecture basically require less mathematical operations to achieve the results;

It should be emphasized, that it was assumed that the "analytical method" leads to the correct results ("true values"). The application of the third architecture together with measured values of losses can constitute in a method with reasonable precision and adequate processing time.

The third ANN offered a global precision as well as the second one. Although this fact is understandable, while taking into consideration that these neural networks are trained to calculate small losses and large percentual error on an individual loss may not be significant on a large scale. It has also the clear advantage that it does not need the calculation of the consumers and transformers m, s curves.

The present ANN application is a computational proof that a reasonable estimate of the losses in a distribution system can be achieved from the methodology proposed. However, it must be pointed out that the parameters used on the training of the ANN have not been exhaustingly optimized, as that was not the main objective on this work. Therefore, there still are some improvements to do on the accuracy issue.

The Alternative 2 ANN Architecture is even faster than the other two options, because it does not need to calculate the consumer profiles and the aggregation for the distribution transformers. Another advantage besides the calculation speed is that the utility does not need to perform measurements to evaluate the load profile of all types of consumers, which is a costly operation.

This methodology shall be incorporated to a Geographical Information System (GIS) so as to turn its calculation procedure more independent from the user interaction.

References

[1] J. A. Jardini, C. M. V. Tahan, S. U. Ahn and S. L. S. Cabral, "Determination of the typical daily load curve for residential area based on field measurements". In: IEEE T&D, 1994, Chicago. v. 2. p. 1-5.

[2] J. A. Jardini, C. M. V. Tahan, S. U. Ahn, R. P. Casolari and F. M. Figueiredo, "Daily load curves - data base established on field measurements". In: International Conference and Exhibition on Electricity Distribution, 1996, Buenos Aires.

[3] A. G. Leal, J. A. Jardini, L. C. Magrini, S. U. Ahn and D. Battani, "Management System of Distribution Transformer Loading". In: Transmission and Distribution 2002, São Paulo, Brazil.

[4] J. A. Jardini, H. P. Schmidt, C. M. V. Tahan, C. C. B. Oliveira and S. U. Ahn, "Distribution transformer loss of life evaluation: a novel approach based on daily load profiles". IEEE Transactions on Power Delivery, United States, v. 15, n. 1, p. 361-366, 2000.

[5] A. G. Leal, "An Information System for the Determination of Losses in Distribution Networks using Typical Demand Curves of Consumers and Artificial Neural Networks". Doctoral Thesis. 158 pages. Polytechnic School, University of Sao Paulo, 2006. (In Portuguese). Brazil.

[6] A. G. Leal, J. A. Jardini, L. C. Magrini, S. U. Ahn, H. P. Schmidt and R. P. Casolari, "Distribution System Losses Evaluation by ANN Approach". In: 2006 IEEE PES Power Systems Conference & Exposition, 2006, IEEE, Atlanta - Georgia, USA.

[7] S. U. Ahn, H. P. Schmidt and D. Battani, "Fast evaluation of technical losses: the concept of equivalent current". In: International Conference on Electricity Distribution, 2003, Barcelona, Spain.

[8] B. C. Degeneff, "Power Transformers". Elsevier Academic Press, The Electrical Engineering Handbook, Chapter on Electric Power Systems, Wai-Kai Chen, Editor, pp. 715-720, Elsevier Academic Press, 2004.

[9] Copper Development Association. “Introduction to Transformer Losses”. In: Premium-Efficiency Motors and Transformers, CDA [Online], V.1, 2002. Available: http://www.copper.org.

7. Biographies

Adriano Galindo Leal, Ph.D. was born in São Paulo, Brazil, on September 19th, 1971. In 1996, he received the B.Sc. Degree in Electrical Engineering from Polytechnic School at University of Sao Paulo. From the same institution, he received the M.Sc. and Ph.D. degrees in 1999 and 2006, respectively. For 11 years, he worked as a R&D Engineer for the GAGTD research group in the Polytechnic School at University of Sao Paulo, where was responsible for the study and development of automation and information systems in the fields of generation, transmission, and distribution of electricity. Since April 2007, he is a Consultant Engineer for Elucid Solutions, a TI company for several Utilities companies in Brazil. His main research interests are Information Systems, Project Management, PMI, Remote Terminal Units, GIS and Artificial Intelligent solutions in the operation of electrical power systems.

José Antonio Jardini, Ph.D. was born in São Paulo, Brazil, on March 27th, 1941. He graduated from the Polytechnic School at University of Sao Paulo in 1963 (Electrical Engineering). From the same institution he received the M.Sc. and Ph.D. degrees in 1971 and 1973, respectively. For 25 years he worked at Themag Engenharia Ltda., a leading consulting company in Brazil, where he jointly guided many power systems studies and participated in major power system projects such as the Itaipu hydroelectric plant. He is currently Professor in the Polytechnic School at Sao Paulo University, where he teaches power system analysis and digital automation. There he also leads the GAGTD group, which is responsible for the research and development of automation systems in the areas of generation, transmission and distribution of electricity. He represented Brazil in the SC-38 of CIGRÉ and is a Distinguished Lecturer of IAS/IEEE.

Se Un Ahn, Ph.D. was born in Inchon, South Korea in 1957. He received his B.Sc. degree from the Mackenzie Engineering School (São Paulo) in 1981, his M.Sc. and Ph.D. degrees in Electrical Engineering from the Polytechnic School at the University of São Paulo in 1993 and 1997, respectively. He works since 1986 as a research engineer in distribution systems at the Piratininga CPFL company (former Eletropaulo and Bandeirantes), all of them being power concessionaires. His professional activities include: load curves use of expansion planning of the electric system.

Advances in Technological Applications of Logical and Intelligent Systems
G. Lambert-Torres et al. (Eds.)
IOS Press, 2009

Author Index

This page intentionally left blank